



**FACULTAD DE POSTGRADO  
TRABAJO FINAL DE GRADUACIÓN**

**FACTORES INFLUYENTES EN LA SELECCION DE LENGUAJES  
DE PROGRAMACION EN EL DESARROLLO WEB EN  
HONDURAS**

**SUSTENTADO POR:**

**HECTOR DAVID SANCHEZ MEJIA**

**PREVIA INVESTIDURA AL TÍTULO DE**

**MÁSTER EN  
GESTION DE TECNOLOGIAS DE LA INFORMACION**

**SAN PEDRO SULA, CORTES, HONDURAS, C.A.**

**Mayo, 2025**

**UNIVERSIDAD TECNOLÓGICA CENTROAMERICANA  
UNITEC**

**FACULTAD DE POSTGRADO**

**AUTORIDADES UNIVERSITARIAS**

**RECTORA**

**ROSALPINA RODRÍGUEZ**

**VICERRECTOR ACADÉMICO NACIONAL**

**JAVIER ABRAHAM SALGADO LEZAMA**

**SECRETARIO GENERAL ROGER**

**MARTÍNEZ MIRALDA**

**DECANA FACULTAD DE POSTGRADO**

**ANA DEL CARMEN RETTALLY VARGAS**

**FACTORES INFLUYENTES EN LA SELECCION DE LENGUAJES  
DE PROGRAMACION EN EL DESARROLLO WEB EN  
HONDURAS**

**TRABAJO PRESENTADO EN CUMPLIMIENTO DE LOS  
REQUISITOS EXIGIDOS PARA OPTAR AL TÍTULO DE  
MÁSTER EN**

**GESTION DE TECNOLOGIAS DE LA INFORMACION**

**ASESOR**

**JESUS RICARDO RODRIGUEZ**

**MIEMBROS DE LA TERNA:**

**ALBA GABRIELA GARAY ROMERO  
CARLOS ROBERTO AMADOR  
DAVID ANTONIO DIAZ GIRON**



## **FACULTAD DE POSTGRADO**

# **FACTORES INFLUYENTES EN LA SELECCION DE LENGUAJES DE PROGRAMACION EN EL DESARROLLO WEB EN HONDURAS**

**Hector David Sánchez Mejía**

### **Resumen**

En este estudio, se presenta un diagnóstico integral sobre los factores que influyen en la elección de lenguajes de programación, abordando aspectos clave como el rubro empresarial, los factores técnicos y la influencia de marcos de trabajo reconocidos internacionalmente, como ITIL. Este análisis no solo identifica los criterios que los equipos utilizan para seleccionar un lenguaje de programación, sino que también propone una rúbrica diseñada para facilitar y estandarizar este proceso en los equipos de desarrollo de software a nivel empresarial. La rúbrica tiene como objetivo mejorar la toma de decisiones, optimizar la alineación con los objetivos del negocio y garantizar una mayor eficiencia en la implementación de soluciones tecnológicas.

**Palabras claves: ITIL, programación, lenguaje, desarrollo, web**



**GRADUATE SCHOOL**

**INFLUENTIAL FACTORS IN THE SELECTION OF  
PROGRAMMING LANGUAGES FOR WEB  
DEVELOPMENT IN HONDURAS**

**Hector David Sanchez Mejia**

**Abstract**

In this study, a comprehensive diagnosis is presented on the factors influencing the choice of programming languages, addressing key aspects such as the business sector, technical factors, and the influence of internationally recognized frameworks like ITIL. This analysis not only identifies the criteria teams use to select a programming language but also proposes a rubric designed to facilitate and standardize this process within corporate software development teams. The rubric aims to improve decision-making, optimize alignment with business objectives, and ensure greater efficiency in the implementation of technological solutions.

**Keywords: ITIL, programming, language, development, web**

## **DEDICATORIA**

A mi familia por su impulso y apoyo constante a lo largo de este viaje académico y por motivarme a persistir en este proceso cuando sentía que dejaba de tener la motivación y la fuerza necesaria para terminar mi postgrado.

## **AGRADECIMIENTO**

A mi padre Hector Noel Sánchez Herrera.

A mi madre Rosa Mélida Mejía Alvarenga.

A mi hermano Noel Eduardo Sánchez Mejía.

A mi tío José Mejía Alvarenga.

## ÍNDICE DE CONTENIDO

DEDICATORIA .....	ix
AGRADECIMIENTO .....	x
ÍNDICE DE CONTENIDO .....	xi
ÍNDICE DE TABLAS .....	xv
ÍNDICE DE ILUSTRACIONES .....	xvi
CAPÍTULO I. PLANTEAMIENTO DE LA INVESTIGACIÓN .....	18
1.1 Introducción .....	18
1.2 Antecedentes del Problema .....	19
1.2.1 Investigación realizada en República Dominicana .....	19
1.2.2 Investigación realizada en México.....	20
1.2.3 Libro guía a la galaxia de aplicaciones móviles .....	21
1.3 Planteamiento del Problema.....	22
1.4 Preguntas de Investigación.....	24
1.4.1 Pregunta General .....	24
1.4.2 Preguntas Específicas .....	24
1.5 Objetivos .....	24
1.5.1 Objetivo General .....	24
1.5.2 Objetivos Específicos .....	24
1.6 Justificación.....	25
CAPÍTULO II – MARCO TEÓRICO .....	27
2.1 Macroentorno .....	27
2.1.1 Investigación relevante en España .....	27
2.1.2 La popularidad de Hypertext Pre-Processor PHP en Francia.....	28
2.1.3 La popularidad de Java en China .....	31
2.2 Microentorno.....	32
2.2.1 Lenguajes de Programación Guatemala.....	32
2.2.2 Exportación de servicios vía tecnología Costa Rica .....	33
2.2.3 Honduras y los factores influyentes .....	34
2.3 Teorías De Sustento .....	36

2.3.1 Cuarta Revolución Industrial.....	36
2.3.2 ITIL .....	38
2.4 Metodologías Temáticas .....	40
2.4.1 Marco de Gestión de Proyectos Ágil Scrum .....	40
2.4.2 Information Technology Infrastructure Library (ITIL).....	41
2.5 Herramientas e Instrumentos.....	42
2.5.1 Marco de Trabajo Ágil Scrum.....	42
2.5.2 Herramientas Tecnológicas .....	46
2.5.3 Instrumentos de recolección de datos.....	47
2.6 Conceptualización .....	47
2.7 Marco Legal .....	49
2.7.1 Marco Legal Internacional .....	49
2.7.2 Marco Legal Nacional .....	50
CAPÍTULO III – METODOLOGIA DE LA INVESTIGACION .....	52
3.1 Tipo de enfoque.....	52
3.2 Alcance de la investigación.....	53
3.3 Diseño de la investigación.....	53
3.3.1 Población .....	54
3.3.2 Muestra.....	54
3.3.3 Técnicas de muestreo .....	54
3.4 Criterios de selección de la muestra.....	55
3.5 Hipótesis.....	56
3.6 Operacionalización de las variables .....	57
3.7 Técnicas, instrumentos, procedimiento y plan de análisis .....	60
3.7.1 Técnicas.....	61
3.7.2 Instrumentos .....	61
3.7.3 Procedimientos .....	61
3.7.4 Plan de análisis .....	62
3.8 Fuentes de Información.....	63
3.8.1 Fuentes Primarias .....	63
3.8.2 Fuentes Secundarias .....	63

3.9 Juicio de expertos .....	63
3.10 Resultados de pruebas piloto.....	64
3.11 Matriz de congruencia.....	65
<b>CAPITULO IV - RESULTADOS Y ANALISIS .....</b>	<b>68</b>
4.1 Análisis Exploratorio de los datos.....	68
4.1.1 Análisis de conocimientos generales.....	68
4.1.2 Análisis sobre ITIL.....	69
4.1.3 Análisis sobre factores técnicos .....	71
4.2 Informe del proceso de recolección de datos .....	76
4.2.1 Descripción del proceso .....	76
4.2.2 Participantes .....	77
4.2.3 Instrumento.....	77
4.2.4 Dificultades encontradas .....	77
4.2.5 Consideraciones éticas .....	77
4.2.6 Generalidades de los datos obtenidos.....	77
4.2.7 Factores técnicos .....	80
4.2.8 Datos obtenidos sobre ITIL.....	84
4.3 Matriz de Correlación .....	90
4.4 Análisis inferencial y modelos aplicados factores determinantes.....	92
4.4.1 Regresión logística .....	92
4.4.2 Naive Bayes.....	93
4.4.3 Random Forest .....	94
<b>CAPITULO V – CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>97</b>
5.1 Conclusiones .....	97
5.2 Recomendaciones .....	97
<b>CAPITULO VI – APLICABILIDAD.....</b>	<b>99</b>
6.1 Instrumento para la elección de lenguajes de programación .....	99
6.2 Justificación de la propuesta .....	99
6.3 Alcance de la propuesta .....	100
6.4 Descripción y desarrollo .....	100
6.4.1 Aplicación práctica a la investigación.....	101

6.4.2	Diseño del instrumento.....	101
6.4.3	Flujograma del proceso de selección de lenguajes.....	102
6.4.3	Guía de uso del instrumento.....	103
6.5	Medidas de control.....	103
6.5.1	Plan de Seguimiento.....	104
6.6	Cronograma de implementación y presupuesto .....	105
6.6.1	Cronograma de implementación en semanas .....	106
6.6.2	Flujograma del proceso de implementación.....	108
6.6.3	Presupuesto del proyecto en Lempiras .....	109
6.7	Matriz de Concordancia.....	110
	REFERENCIAS BIBLIOGRÁFICAS.....	113
	ANEXOS .....	119

## ÍNDICE DE TABLAS

Tabla 1 Comparación entre Python y R.....	21
Tabla 2 Ranking de lenguajes de programación TIOBE .....	31
Tabla 3 Venta de servicios por TIC Costa Rica.....	33
Tabla 4 Cantidad de ingresos por rubro, servicios tecnológicos Costa Rica .....	34
Tabla 5 Ventajas SCRUM .....	43
Tabla 6 Comparativa de Scrum con otras herramientas .....	45
Tabla 7 Comparativa de Herramientas .....	46
Tabla 8 Leyes o Normativas Internacionales.....	50
Tabla 9 Leyes o Normativas Nacionales .....	51
Tabla 10 Tamaño de la Muestra.....	55
Tabla 11 Criterios de Inclusión y Exclusión.....	56
Tabla 12 Operacionalización de variables .....	57
Tabla 13 Plan de análisis.....	62
Tabla 14 Análisis de conocimientos generales .....	69
Tabla 15 Valores nulos .....	70
Tabla 16 Análisis ITIL.....	71
Tabla 17 Análisis factores técnicos.....	72
Tabla 18 Factores elegidos mayor a menor .....	75
Tabla 19 Regresión logística.....	93
Tabla 20 Naive Bayes .....	94
Tabla 21 Random Forest.....	95
Tabla 22 Resumen modelos aplicados.....	96
Tabla 23 Instrumento para la elección de lenguajes de programación .....	101
Tabla 24 Plan de seguimiento .....	104
Tabla 25 Cronograma de implementación en semanas.....	106
Tabla 26 Presupuesto en Lempiras .....	109
Tabla 27 Matriz de Concordancia.....	110
Tabla 28 Instrumento Realizado .....	119
Tabla 29 Juicio de Expertos .....	126

## ÍNDICE DE ILUSTRACIONES

Ilustración 1 Lenguajes de Programación Guatemala .....	32
Ilustración 2 Revoluciones Industriales .....	37
Ilustración 3 Lenguajes de Programación en IoT .....	38
Ilustración 4 Ciclo de vida del desarrollo de software según ITIL.....	39
Ilustración 5 Cadena de Valor del Servicio ITIL.....	41
Ilustración 6 Estructura conocimientos generales.....	68
Ilustración 7 Estructura tratamiento de datos ITIL .....	70
Ilustración 8 Estructura factores técnicos .....	71
Ilustración 9 Estructura factores técnicos .....	72
Ilustración 10 Factores con mayor media .....	76
Ilustración 11 Rol Profesional .....	77
Ilustración 12 Años de Experiencia Profesional.....	78
Ilustración 13 Conocimiento de ITIL .....	79
Ilustración 14 Conocimiento sobre metodologías ágiles para la gestión de proyectos.....	80
Ilustración 15 Factores considerados claves para elegir lenguajes de programación .....	81
Ilustración 16 Considera la experiencia un factor para tomar en cuenta .....	82
Ilustración 17 Considera importante la abundancia de librerías .....	82
Ilustración 18 Considera importante el rubro de la empresa .....	83
Ilustración 19 Factores técnicos importantes para elegir un lenguaje .....	84
Ilustración 20 Participación en proyectos influenciados ITIL.....	84
Ilustración 21 Faceta de ITIL de mayor relevancia al elegir un lenguaje.....	85
Ilustración 22 Alineación lenguajes con prácticas de ITIL .....	85
Ilustración 23 Influencia de ITIL en la empresa.....	86
Ilustración 24 Porcentaje de participación en proyectos que se complican por la elección de un lenguaje .....	87
Ilustración 25 Porcentaje de consideración de costos asociados (infraestructura, licencias) .....	88
Ilustración 26 Aceptación de rúbrica .....	88
Ilustración 27 Mapa de calor de criterios prioritarios.....	89
Ilustración 28 Mapa de calor matriz de correlación.....	91
Ilustración 29 Estructura regresión logística.....	92

Ilustración 30 Estructura Naive Bayes.....	93
Ilustración 31 Estructura Random Forest .....	95
Ilustración 32 Flujograma del proceso de selección de lenguajes .....	102
Ilustración 33 Flujograma del proceso de implementación .....	108

# CAPÍTULO I. PLANTEAMIENTO DE LA INVESTIGACIÓN

## 1.1 Introducción

El mundo y la humanidad se encuentran inmersos en una transformación digital constantemente, donde la tecnología evoluciona a pasos agigantados. Los lenguajes de programación desempeñan un papel relevante y crucial en el progreso de nuestro entorno.

Desde la creación de las primeras computadoras, los lenguajes de programación han evolucionado desde poder ejecutar simples instrucciones a sofisticadas herramientas que nos permiten interactuar con máquinas complejas. Esta evolución ha sido paralela al desarrollo de la sociedad, dando origen a internet, la inteligencia artificial, la realidad virtual y una infinidad de aplicaciones que han transformado nuestra forma de vivir y trabajar.

Una prueba de ello es lo recalcado en el estudio realizado por (CEPAL, 2021) donde afirma que la incursión de la tecnología en el comercio ha supuesto un importante crecimiento, se estima que en el año 2019 un total de 1,920 millones de personas compraron productos y adquirieron servicios en línea y que el total de ventas generadas en este rubro superaron los 3,5 billones de dólares en todo el mundo. Estos abrumadores números no serían posibles sin la existencia de plataformas digitales que propician el comercio electrónico, tales como Amazon, PedidosYa, eBay, Shopify, Mercado Libre, etc.

Este auge tecnológico implica la creación de muchos lenguajes de programación y de frameworks que heredan en gran parte su lenguaje base, donde cada lenguaje busca mejorarse uno al otro y propicia que el mercado se encuentre saturado.

El capítulo I hace énfasis en la explicación del motivo para llevar a cabo la investigación, toma en consideración investigaciones realizadas en otros lugares y como enfrentaron esta problemática. Además, se plasma la justificación de la investigación y el motivo para realizarla.

El capítulo II contiene el marco teórico donde se analizan elementos como el macroentorno, microentorno y se mencionan las teorías de sustento y las herramientas temáticas para llevar a cabo la investigación.

El capítulo III se basa en la metodología de la investigación, contiene elementos claves de la investigación tales como el enfoque, alcance y diseño. En cada una de ellas se brindará una

explicación de las razones que justifican su elección por encima de otras opciones.

Agregado a lo expuesto anteriormente, este capítulo contiene elementos como la población, muestra y los instrumentos necesarios para recolectar datos. También se dejan plasmados los criterios a considerar, así como también se deja plasmado el procedimiento y posterior análisis que se llevaran a cabo una vez se cuente con los datos para comenzar con su interpretación.

El capítulo IV muestra el análisis de los datos y ciertos resultados obtenidos que fueron obtenidos luego de la aplicación del instrumento realizado. En este capítulo además se muestra una matriz de correlación la cual será importante para evaluar los datos y cumplir con el objeto del estudio.

El capítulo V tiene como objetivo presentar las conclusiones y recomendaciones de la investigación, las cuales estarán fundamentadas luego de analizar el conjunto de datos utilizando modelos estadísticos.

En el capítulo VI se muestra la aplicabilidad del proyecto y la propuesta de este, dicha propuesta se basa en el análisis de los datos estudiados en el capítulo IV.

## **1.2 Antecedentes del Problema**

A pesar de la cantidad de lenguajes de alto nivel que existen hoy en día, entiendo que es necesario facilitar una rúbrica o marco de referencia que pueda proporcionar los pasos o fases a seguir para elegir un lenguaje y que el elegido sea el adecuado para nuestro entorno y sus variantes.

La cantidad de lenguajes que hay hoy en día provoca desorganización internamente en las empresas al tener un abanico de posibilidades tecnológico amplio, obligándolos a desarrollar en diferentes lenguajes.

### **1.2.1 Investigación realizada en República Dominicana**

Prueba de ello es la investigación de tesis realizada en la Universidad APEC por los autores Alan Rodríguez y Luis Florentino, su trabajo tiene por nombre “Análisis E Implementación De La Programación De Software Como Asignatura Básica En Las Escuelas Públicas Y Privadas A Nivel De Educación Media, En La Ciudad De Santo Domingo Durante El Periodo septiembre-diciembre 2020. Caso De Estudio: Instituto Tecnológico Fabio Amable Mota”, en el cual ellos definen un lenguaje de programación de la siguiente forma: “Un lenguaje de programación se define como una serie de instrucciones que usan los programadores para crear programas de software ya sean

aplicaciones móviles, web o de escritorio” (Rodríguez & Florentino, 2020).

De esta forma, ambos autores nos exponen que la programación no simplemente es basada y destinada a un dispositivo objetivo y que esta gran variedad de dispositivos en parte promueve el desarrollo de lenguajes para desarrollar a estas nuevas plataformas emergentes.

Según sea la finalidad del proyecto ciertos lenguajes y paradigmas pueden ser utilizados para obtener un mejor beneficio en el futuro. En este aspecto, es importante considerar si el proyecto requiere funcionalidades e integraciones con terceros, por ejemplo, si se necesita integrar el servicio de mensajería en una aplicación móvil es muy probable que se seleccione Google como proveedor de este servicio, el cual tiene ciertos requisitos que deben ser cumplidos, tales como:

- Versión del Lenguaje de Programación.
- Dispositivo objetivo (Android, iOS, Web, Escritorio).
- Sistema Operativo base.

### **1.2.2 Investigación realizada en México**

Se identificó un segundo trabajo de tesis realizado por su autor Ricardo López en el Tecnológico Nacional de México, el nombre de la investigación es: “Análisis Comparativo De Lenguajes De Programación Para El Desarrollo De Aplicaciones En Ciencia De Datos” si bien es cierto, el trabajo investigativo no es basado directamente sobre el desarrollo de software específicamente, el autor expone y señala que los lenguajes de programación son empleados para desarrollar aplicaciones en múltiples campos de conocimiento, sin embargo, en la Ciencia de Datos un lenguaje es clave para efectuar tareas como la lectura, exploración, modelado y la visualización de datos. Por esta razón, no todos los lenguajes son funcionales y adecuados para ser utilizados en el ámbito de Ciencia de Datos (López, 2020).

Esta investigación tiene como finalidad determinar el lenguaje de programación que mejor responda a actividades de Ciencias de Datos, durante el desarrollo del informe se muestra comparaciones relacionadas entre lenguajes según criterios establecidos. Se recalca la importancia del hardware y su influencia en la programación, ya que no es lo mismo ejecutar un programa en un computador con procesador Intel Pentium que ejecutarlo en uno con procesador Intel Core 7.

Desde esta perspectiva, se deja patente que la variedad de lenguajes de programación no

significa que todos puedan ser utilizados para cualquier tipo de tarea, cada uno presenta sus propias características, sus ventajas y desventajas.

Dentro del trabajo se realizaron varias comparaciones principalmente entre dos lenguajes, Python y R. El señor López manipulo en total 10 base de datos y obtuvo los siguientes resultados:

**Tabla 1 Comparación entre Python y R**

Nombre de la base de datos	Número de objetos	d	Lenguaje ganador en tiempo	Lenguaje ganador en calidad	% de pérdida de calidad con R	% de pérdida de calidad con Python	% de ganancia de tiempo con R	% de ganancia de tiempo con Python
Bank	45211	6	R	R	-----	0.00980121	52.3469715	-----
Breastcancer	683	9	R	Python	0.12300381	-----	68.752212	-----
Iris	150	4	R	Python	14.2891217	-----	<b>99.5056322</b>	-----
Poker-hand-testing	1000000	11	Python	R	-----	0.0144545	-----	<b>40.0369797</b>
Powerplant	9568	5	R	R	-----	0.00204643	47.9540322	-----
Rssi	1420	13	R	R	-----	<b>9.38E-07</b>	69.4100395	-----
Skin_NonSkin	245057	4	R	R	-----	0.0142223	54.1074337	-----
Wine	178	13	R	Python	0.00000666	-----	68.67392	-----
Household	2049280	4	R	R	-----	0.00747702	60.9984204	-----
3D_spatial_network	434874	4	R	R	-----	<b>1.71E-05</b>	63.5796806	-----

Fuente: López. R (2020)

### 1.2.3 Libro guía a la galaxia de aplicaciones móviles

Este libro realizado por múltiples actores se destaca por su enfoque en las aplicaciones móviles y trata de profundizar sobre las alternativas y proponer una hoja de ruta para aquellos que desean involucrarse en este nicho de la programación.

Sobre este tipo de desarrollo, el autor define que las aplicaciones nativas brindan un rendimiento superior, una integración más completa y una experiencia de usuario optimizada en comparación con otras alternativas. No obstante, el desarrollo nativo suele ser la opción más compleja de implementar Robert Virkus (2016).

Hablando en términos de rendimiento una aplicación nativa sobresale en comparación del

desarrollo multiplataforma Robert Virkus (2016). Sin embargo, y por experiencia propia puedo resaltar que el desarrollo multiplataforma facilita el mantenimiento y escalabilidad de esta. Siempre y cuando la aplicación no requiera funcionalidades propias del sistema operativo (Android, iOS) considero que debe tomarse en cuenta esta alternativa por sobre el desarrollo nativo.

El auge de las aplicaciones móviles aún se encuentra latente en el mercado, prueba de ello es el estudio realizado por (DATA.AI, 2024) el cual es un sitio especializado en el nicho del desarrollo móvil donde en su más reciente estudio revelan que el gasto de usuarios se aproxima a 171 mil millones de dólares lo que supone un incremento del 3% interanual, se estima que el total de descargas de aplicaciones móviles fueron 257 mil millones y las horas dedicadas en todo el mundo al uso de una aplicación móvil alcanzaron el máximo de 5,1 billones. Cabe recalcar que estos datos incluyen información de las tiendas de todo el mundo y también de las más famosas como PlayStore de Google y AppStore de Apple.

### **1.3 Planteamiento del Problema**

En la actualidad, el avance tecnológico ha propiciado el surgimiento de una gran variedad de lenguajes de programación. Esta abundancia de opciones, lejos de simplificar el proceso de selección, aumenta la complejidad para los encargados de desarrollo en una empresa y para desarrolladores aficionados, quienes deben destinar un tiempo considerable a evaluar cuál lenguaje se adapta mejor a los requerimientos específicos del proyecto. A esto se suma la existencia de frameworks, que son herramientas elaboradas sobre un lenguaje de programación, las cuales buscan optimizar su uso, mejorar su rendimiento y resolver problemáticas que el lenguaje base presenta. Estos frameworks, aunque heredan los fundamentos de su "lenguaje padre", aportan características adicionales y hasta sintaxis diferente creando una mayor diversidad de alternativas, pero también más decisiones por tomar.

Por ejemplo, en JavaScript se cuentan con frameworks que son mundialmente usados y reconocidos, cada uno cuenta con su propia comunidad que promueve su uso y se encargan de brindar soluciones que faciliten la integración. Para este ejemplo tomemos únicamente a 2 frameworks. Angular y React, el primero (Angular) es propio de Google y el segundo (React) fue hecho por Facebook, de este modo ambos cuentan con un prestigio que promueve su uso por sí solos sin la necesidad de pagar para utilizarlo (React Community, 2024).

La problemática se intensifica cuando consideramos que:

- **El tiempo de evaluación se ha multiplicado:** Los desarrolladores deben analizar múltiples dimensiones (rendimiento, escalabilidad, curva de aprendizaje) antes de iniciar un proyecto.
- **Los frameworks añaden capas de complejidad:** Estas herramientas, construidas sobre lenguajes base, introducen nuevas variables de decisión. Por ejemplo:
  - Mientras JavaScript como lenguaje base tiene un comportamiento predecible, frameworks como Angular o React modifican sustancialmente los flujos de trabajo.
  - Cada framework trae su propio ecosistema de herramientas complementarias (state managers, SSR solutions) que deben evaluarse en conjunto.
- **Diversidad entre los ecosistemas de los frameworks:** Esta diversidad genera un fenómeno de "fatiga tecnológica" debido a la saturación de lenguajes disponibles en el mercado y la necesidad de estar al tanto de cambios.
- **Costos Ocultos:** Las empresas enfrentan costos ocultos por:
  - Capacitación continua.
  - Riesgo de obsolescencia tecnológica.
  - Dificultad para encontrar talento especializado en stacks específicos.
- **Evolución Constante:** La evolución paralela de lenguajes y frameworks crea un desafío de sincronización:
  - Los lenguajes base se actualizan.
  - Los frameworks deben adaptarse a estos cambios.
  - Los desarrolladores deben mantenerse actualizados en apartados simultáneamente.

Los aspectos indicados anteriormente reflejan una complejidad multidimensional que da indicios de atrasos en proyectos de software. Debido a esto surgen una serie de herramientas como StackShare y BestOfJs que se enfocan a ayudar en la toma de decisiones mediante comparativas técnicas detalladas.

Por ende, la selección de un lenguaje de programación, lejos de ser una tarea puramente técnica, es un paso determinante que tendrá implicaciones directas para el proyecto y para su

mantenibilidad en el futuro. La mayor complejidad es encontrar el punto medio entre varios factores: las necesidades técnicas del proyecto, las habilidades y experiencia del equipo de desarrollo, y la integración con otros sistemas que ya existen en la empresa. Muchas veces, el lenguaje más familiar para el equipo no es el más adecuado para el proyecto en términos de rendimiento o escalabilidad. Esto obliga a los desarrolladores a tomar decisiones estratégicas que maximicen el uso de sus recursos, pero sin comprometer la calidad y sostenibilidad del proyecto.

## **1.4 Preguntas de Investigación**

### **1.4.1 Pregunta General**

¿Qué factores, criterios técnicos y contextuales influyen en las preferencias de los profesionales del software al seleccionar lenguajes de programación para el desarrollo web, y cómo se manifiestan estas preferencias según su experiencia y rol?

### **1.4.2 Preguntas Específicas**

1. ¿Cuáles son los 5 factores más influyentes en la selección de un lenguaje de programación entre profesionales del desarrollo de software?
2. ¿Qué criterios pueden utilizarse para construir una rúbrica que facilite la selección de lenguajes de programación?

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Diagnosticar las preferencias de los profesionales del software en el contexto del desarrollo web para comprender los criterios que guían la elección de lenguajes de programación, con el propósito de identificar factores determinantes en dicha selección.

### **1.5.2 Objetivos Específicos**

1. Identificar los 5 principales factores que influyen en la selección de un lenguaje de programación entre profesionales del área de desarrollo de software.
2. Diseñar una rúbrica compuesta de los factores identificados que sirva como herramienta de orientación para los responsables de desarrollo en empresas al momento de seleccionar un lenguaje de programación.

## 1.6 Justificación

El presente trabajo de investigación se realizó con la finalidad de hacer conciencia en la importancia que tiene una correcta selección de un lenguaje de programación. Esta elección no solo impacta al Departamento de IT de una empresa sino también a la empresa en general y puede resultar contraproducente para la misma.

Según un estudio realizado en España, precisamente por la Escuela Superior de Informática, se estima que hay alrededor de 600 lenguajes de programación (Agency, 2021), lo cual recalca el continuo avance de la tecnología y la diversificación de los lenguajes generando un gran abanico de posibilidades, por ende, los equipos de desarrollo ya no solo deben tomar en consideración elementos puramente técnicos.

Un ejemplo para recalcar la importancia de la elección y los factores a tomar en cuenta para esta decisión ocurrió en el año 2014, donde un grupo de personas malintencionadas detectaron y aprovecharon una vulnerabilidad llamada “Heartbleed” en el lenguaje de programación C, la empresa afectada fue la mundialmente conocida OpenSSL la cual es muy utilizada por múltiples empresas en sus servidores permitiendo a los atacantes leer información incluyendo claves privadas y datos sensibles (HEARTBLEED, 2018) .

Otro caso relevante se produjo en la empresa Knight Capital la cual era una empresa de servicios financieros que operaba directamente en Wall Street. Todo marchaba con normalidad cuando de repente un error en el software que era encargado de ejecutar las acciones de la empresa ejecutó de forma no controlada las transacciones que se tenían planeadas para toda una semana. Este error o bug estuvo presente durante 45 minutos, según estadísticas por cada minuto la empresa perdía alrededor de 10 millones de dólares, la pérdida total causada ascendió hasta los 440 millones de dólares dejando a la empresa prácticamente en quiebra (Otto, 2017).

Posteriormente, la empresa busco inversores para levantarse financieramente y fue cuando uno de sus máximos competidores compró el 100% de la compañía.

En este contexto, la investigación pretende hacer constar el impacto que puede tener un lenguaje de programación en el ciclo de vida del proyecto y su relación con las practicas plasmadas y sugeridas por el marco ITIL v4. Entre las cuales se destacan las practicas (AXELOS, 2019):

- Gestión de Rendimiento: Un lenguaje que optimice procesos puede permitir mejorar la

carga de trabajo de la infraestructura y responda de mejor forma a las necesidades del cliente aportando valor.

- **Gestión de la Demanda:** A medida una empresa crece se hace vital mejorar los servicios de TI para que puedan abastecer las solicitudes de clientes. Un lenguaje de programación que permita una escalabilidad sin complejos puede resultar vital.
- **Gestión de Activos:** El software final pasará a ser un activo de TI, escoger un lenguaje que permita la actualización de librerías propias y la posibilidad de emprender nuevos proyectos con el mismo lenguaje resalta la importancia de la elección.
- **Gestión del Conocimiento:** Un recurso de TI que cuente con amplia documentación, una comunidad activa e incluso soporte de parte del creador del lenguaje calza perfectamente con esta práctica que busca garantizar el acceso a la información a todo el equipo y no que únicamente ciertos integrantes tengan conocimiento.

Sin embargo, a pesar de que en ciertos lenguajes existe una gran cantidad de información, muy pocos documentos proponen una guía que sirva de base a aquellos que tengan que seleccionar un lenguaje de programación entre tantos.

## **CAPÍTULO II – MARCO TEÓRICO**

La historia de la humanidad ha estado marcada por constantes cambios y transformaciones que han moldeado nuestra realidad actual. En los últimos siglos, la tecnología ha desempeñado un papel fundamental en este proceso, impactando profundamente todos los ámbitos de la vida humana. Gracias a los avances tecnológicos, se ha logrado potenciar la capacidad de producción, acelerar el desarrollo económico y social, y expandir los límites del pensamiento humano, impulsando así una evolución sin precedentes en nuestra forma de vivir, crear y comprender el mundo.

Pensar en el mundo de hoy sin todos los beneficios que tenemos se hace difícil y es aquí donde la programación y el desarrollo de software han jugado un papel relevante en la sociedad y sus diferentes rubros. La programación aparte de facilitar la creación de nuevas tecnologías, fomenta y promueve la innovación en el mercado global (Universidad de Galileo, 2024)

La historia de la humanidad ha sido un constante proceso de cambio y transformación, donde la tecnología ha desempeñado un papel clave en la evolución social y económica. En las últimas décadas, herramientas como la programación, en conjunto con otras tecnologías, han revolucionado la manera en que las personas se comunican e intercambian información. Aunque el Internet es el núcleo de esta interconexión global, su potencial sería limitado sin navegadores, aplicaciones y dispositivos diseñados para facilitar el acceso a este vasto ecosistema digital.

Un ejemplo concreto de esta integración tecnológica se observa en el sector turístico. Según un estudio realizado en México, las Tecnologías de la Información han demostrado ser un aliado estratégico para promocionar destinos turísticos, funcionando como una vitrina que impulsa el atractivo de las regiones. En este contexto, las empresas deben adaptarse e implementar soluciones digitales, ya que su público objetivo, principalmente jóvenes altamente conectados, depende de estas herramientas para planificar y enriquecer sus experiencias de viaje. Así, la tecnología no solo transforma la manera en que vivimos, sino también cómo exploramos y comprendemos el mundo (Esteban, 2022).

### **2.1 Macroentorno**

#### **2.1.1 Investigación relevante en España**

De igual forma, se encontró una investigación de tesis en España, precisamente en Barcelona. El trabajo tiene por nombre: “Análisis De Rendimiento De Lenguajes De Desarrollo

De Aplicaciones Y Servicios Web” este trabajo fue realizado por Víctor Gallego. Víctor propone un estudio teórico y práctica para evaluar el rendimiento de ciertos lenguajes de programación. En el trabajo se encuentra la siguiente descripción: “Un paradigma de programación es un conjunto de técnicas y principios que se utilizan para diseñar y desarrollar software. Cada paradigma tiene sus propias fortalezas y debilidades, y se adapta mejor a ciertos tipos de problemas” (Gallego, 2023, p. 38).

A partir de este análisis, se concluye que la programación no se limita a un único enfoque, sino que engloba múltiples paradigmas sobre los cuales se basan y enfocan los distintos lenguajes de programación. Cada paradigma resulta especialmente útil dependiendo del problema a resolver, las necesidades del proyecto y las características del entorno en el que se desarrolla. Por ejemplo, algunos paradigmas son más adecuados para resolver problemas específicos de estructura de datos, mientras que otros se enfocan en la concurrencia, la orientación a objetos o el procesamiento funcional. La elección del paradigma correcto es clave para lograr soluciones eficientes y optimizadas en términos de rendimiento, mantenimiento y escalabilidad del software.

Además, en el mismo estudio se aborda la relevancia de los frameworks en el desarrollo de aplicaciones web, destacando las ventajas y desventajas que su uso conlleva. Los frameworks son herramientas que facilitan y agilizan el desarrollo al proporcionar estructuras predefinidas y componentes reutilizables, pero su implementación también debe ser cuidadosamente evaluada. Gallego menciona los siguientes puntos clave:

En el mismo documento, se hace referencia a la inclusión de un framework al momento de desarrollar aplicaciones web y sus ventajas y desventajas:

- Fiabilidad en el código
- Estándares y mejores prácticas.
- Agrega una capa de abstracción, que puede suponer afectaciones en el rendimiento según sea el uso.
- Librerías utilizadas que pueden ser deprecadas dejando inoperante el software.

### **2.1.2 La popularidad de Hypertext Pre-Processor PHP en Francia**

Hablar de tecnología generalmente se relaciona con economía, muchos proyectos tecnológicos demandan una inversión económica que esta solo al alcance de países fuertes

económicamente. Estas inversiones generalmente son las que hacen que un país se siga manteniendo como punto de referencia ante los demás países del mundo.

Este es el caso de Francia, un país que no tiene nada que envidiarles a las demás potencias del mundo, se estima que a principios de 2024 Francia figuraba entre las 10 economías más importantes (The Top 25 Economies in the World, 2024).

Hablando puramente de tecnología e innovación, Francia se ubica en la posición número 12 a nivel mundial de los países que trazan la ruta a seguir sobre este apartado respecto a los demás países, lo cual demuestra la firme creencia en el desarrollo de nuevos productos y la búsqueda constante de la innovación (*Índice Mundial de Innovación de 2024*, 2024).

Francia es un país que ha adoptado a PHP como su lenguaje insignia y, sin embargo, PHP no se caracteriza por ser un lenguaje de programación que lidere en áreas de innovación tecnológica como la inteligencia artificial, la ciencia de datos o el Internet de las cosas. Su diseño y propósito original, centrados principalmente en el desarrollo web, lo han mantenido relevante en este ámbito, pero lo han dejado rezagado frente a lenguajes más versátiles y modernos que dominan en sectores emergentes. De acuerdo con el índice de popularidad de lenguajes de programación de TIOBE, PHP ocupa actualmente el puesto número 12 a nivel mundial, lo que refleja una sólida base de usuarios, aunque su presencia en áreas de tecnología de punta es limitada en comparación con otros lenguajes más especializados o recientes (*TIOBE Index*, 2024).

Definitivamente, los datos presentados no hacen alusión a que Francia siendo la potencia económica y tecnológica que demuestra ser utilice mayormente el lenguaje de programación PHP por delante de otros lenguajes considerados vanguardistas ya que se estima que la popularidad de PHP entre los desarrolladores en Francia es del 43% siendo utilizado como lenguaje principal (JetBrains, 2021b). Para entender el motivo de este suceso se plantean una serie de aspectos:

- **Facilidad de aprendizaje:** PHP es un lenguaje de programación sencillo de aprender y esto repercute en que sea ampliamente utilizado. En el caso de Francia y de otros países de la región europea se comienza su aprendizaje desde un nivel educativo muy básico, lo cual hace que ingenieros de software profesionales cuenten con un amplio conocimiento del lenguaje y tengan vasta experiencia en el uso del lenguaje.
- **Comunidad activa:** Ser el país donde mayor popularidad tiene PHP se traduce en que existen

muchos programadores activos del lenguaje que promueven la existencia de una comunidad activa capaz de apoyar en problemas, compartir herramientas y sobre todo mantener al lenguaje vivo y en constante crecimiento.

- **Compatibilidad con Sistema de Gestión de Contenidos CMS:** El auge de las plataformas web propuso un nuevo sistema para el desarrollo de estas, esto fue el inicio de herramientas como WordPress, Joomla o Drupal los cuales están basados y contruidos en PHP, de esta forma facilita su integración con estas plataformas emergentes.
- **Grandes proyectos realizados en el lenguaje:** Francia ha sido el epicentro de desarrollo de muchas plataformas conocidas a nivel mundial y la mayoría de estas plataformas han sido contruidas para sitios webs con el lenguaje PHP. Sin duda esto promueve el uso del lenguaje en la sociedad. Entra las plataformas desarrolladas en Francia con este lenguaje y que son conocidas mundialmente tenemos:
  - Dailymontion.
  - PrestaShop.
  - Symfony.
- **Compatibilidad con Sistema de Gestión de Contenidos CMS:** El auge de las plataformas web propuso un nuevo sistema para el desarrollo de estas, esto fue el inicio de herramientas como WordPress, Joomla o Drupal los cuales están basados y contruidos en PHP, de esta forma facilita su integración con estas plataformas emergentes.
- **Normativa de privacidad de datos:** Europa cuenta con una normativa que obliga proteger los datos de los usuarios que son tomados por las aplicaciones o plataformas. La normativa utilizada es El Reglamento General de Protección de Datos (GDPR), el cual tiene como finalidad contribuir a un espacio libre, seguro y justo donde se respete la integridad de la información personal de las usuarios finales de las distintas plataformas (GDPR UE, 2016).

**Tabla 2 Ranking de lenguajes de programación TIOBE**

Nov 2024	Nov 2023	Change	Programming Language	Ratings	Change
1	1		 Python	22.85%	+8.69%
2	3	▲	 C++	10.64%	+0.29%
3	4	▲	 Java	9.60%	+1.26%
4	2	▼	 C	9.01%	-2.76%
5	5		 C#	4.98%	-2.67%
6	6		 JavaScript	3.71%	+0.50%
7	13	▲	 Go	2.35%	+1.16%
8	12	▲	 Fortran	1.97%	+0.67%
9	8	▼	 Visual Basic	1.95%	-0.15%
10	9	▼	 SQL	1.94%	+0.05%
11	16	▲	 Delphi/Object Pascal	1.48%	+0.33%
12	7	▼	 PHP	1.47%	-0.82%
13	14	▲	 MATLAB	1.28%	+0.12%
14	20	▲	 Rust	1.17%	+0.26%
15	17	▲	 Swift	1.14%	+0.11%

Fuente TIOBE (2024)

### 2.1.3 La popularidad de Java en China

En Asia, precisamente en China, un país denominado pionero en muchos aspectos tanto tecnológicos, comerciales y de producción según un informe de La Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura (UNESCO) en el año 2015 se puso en marcha un programa llamado “Made in China 2025”. El objetivo principal de este programa estaba orientado en reducir la dependencia del país de tecnologías extranjeras fundamentales (UNESCO, 2021)

El reporte también menciona que para el año 2030 China estima convertirse en el país más importante del mundo en el campo de la innovación tecnológica, en el año 2021 China fue considerado el país con mayores patentes de inteligencia artificial, sin embargo, carecía de suficiente personal para realizar investigación en ese ámbito.

Sobre lenguajes de programación, un estudio realizado por JetBrains indica que Java es el lenguaje mayormente utilizado con un 47% sobre otro tipo de tecnologías y lenguajes considerados emergentes y vanguardistas (JetBrains, 2021a). Esto difiere de la apuesta de China en el rubro de la inteligencia artificial, puesto que Java no es comúnmente utilizado para desarrollar sobre esa

tecnología, sin embargo, su tasa de uso puede explicarse sobre el modelo educativo chino y la preferencia a enseñar este lenguaje desde tempranas edades que permite conocer principios de programación indispensables para cualquier tipo de desarrollo, por ejemplo, estructura de datos, tipado, programación orientada a objetos, etc.

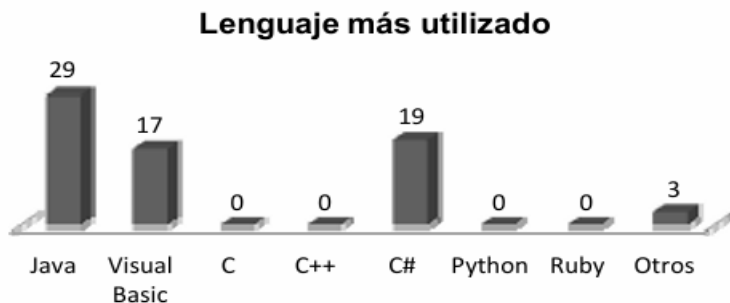
## 2.2 Microentorno

### 2.2.1 Lenguajes de Programación Guatemala

Adentrándonos en Centro América precisamente en Guatemala, un país el cual se encuentra en constante crecimiento económico obteniendo un porcentaje mayor de crecimiento superior a otros países de la región centroamericana y del Caribe. Se estima que en el 2023 Guatemala obtuvo un producto interno bruto (PIB) de 104,4 mil millones de dólares (Banco Mundial, 2024).

Luego de la presentación de los datos económicos de Guatemala, retomamos el tema de interés sobre lenguajes de programación. Siempre en ese mismo país se identificó un trabajo investigativo que recaba datos sobre el uso de lenguajes de programación Cascadas (2011) comparte la ilustración 1.

Esta grafica nos demuestra la preferencia de los programadores por tecnologías conocidas mundialmente y que cuentan con una fama considerable en el desarrollo de soluciones de software tales como Java y C#, llama la atención la nula elección de Python siendo este un lenguaje de tendencia en el mercado y que puede utilizarse para distintos fines como el desarrollo, la ciencia de datos, internet de las cosas, inteligencia artificial, etc.



#### **Ilustración 1 Lenguajes de Programación Guatemala**

Fuente: (Cascadas, 2011).

Estos datos dejan en evidencia que la elección de un lenguaje de programación no suele ir de la mano con aspectos económicos del país, puesto que economías fuertes como Francia y

Guatemala apuestan por lenguajes que no están de moda globalmente.

### 2.2.2 Exportación de servicios vía tecnología Costa Rica

Costa Rica, el país tecnológicamente mejor posicionado de la región cuenta con múltiples avances sobre los demás países de la región. Prueba de ello es la sede que tiene Intel en territorio costarricense.

Intel en Costa Rica significa aproximadamente el 60% de exportaciones en el rubro de investigación y desarrollo en el país se mantienen en constante crecimiento. Este gigante de la computación promueve el desarrollo de la ciencia, tecnología e ingeniería mediante voluntariados y donaciones a programas juveniles. Mas halla del impacto económico que genera, Intel está involucrado estrechamente con las universidades públicas y privadas del país a las cuales apoya en la elaboración de planes de estudios universitarios para mejorar la experiencia educativa del estudiante promoviendo una mejor empleabilidad en el futuro (Intel, 2025).

La influencia de grandes empresas tecnológicas en el país puede confirmarse en estudios económicos del país, precisamente el Banco Central de Costa Rica (BCCR) realizó un estudio sobre la cantidad de dinero generada por medio de servicios tecnológicos. El BCCR presenta que en el año 2022 las exportaciones de este tipo de servicios fueron por un gran total de USD 5539 millones, un monto que significó el 43.4% de las exportaciones de servicios y el 8% del producto interno bruto (Banco Central de Costa Rica, 2022). Lo datos pueden verse en la siguiente tabla.

**Tabla 3 Venta de servicios por TIC Costa Rica**










Tamaño	Número de empresas	Ventas TIC USD Millones
Micro (1)	524	54
Pequeña (2)	58	49
Mediana (3)	129	466
Grande (4)	288	4.970
Total general	999	5.539

Fuente: BCCR (Banco Central de Costa Rica, 2022)

Dentro del informe se presenta una tabla con el rubro que mayor participación tiene en la venta de servicios. Los datos presentados muestran un gran total de ingresos por servicios de informática y software, precisamente de USD 1271 millones; el cual es un monto importante y que

refleja el abanico de oportunidades que puede generar un software construido sobre lenguajes de programación.

**Tabla 4 Cantidad de ingresos por rubro, servicios tecnológicos Costa Rica**

Tipo de servicio	Ventas TIC USD Millones	Participación relativa (%)
 Telecomunicaciones	29	0,5
 Informática y software	1.271	22,9
 Mercadeo y ventas	108	2,0
 Información	17	0,3
 Seguros y financieros	68	1,2
 Administración y oficinas auxiliares	3.794	68,5
 Licencia	6	0,1
 Ing. investigación y desarrollo	241	4,4
 Educación y entrenamiento	5	0,1
<b>TOTAL</b>	<b>5.539</b>	<b>100</b>

Fuente: BCCR (Banco Central de Costa Rica, 2022)

### 2.2.3 Honduras y los factores influyentes

Los elementos planteados en otros países no son los mismos que en Honduras, ya que no se tiene el mismo contexto social ni cultural de Francia, en el año 2022 según investigación realizada por el Instituto Nacional de Estadísticas (INE) solo el 17.1% tiene acceso a una computadora en la casa (INE, 2022), lo cual limita la exposición de los jóvenes a la computación desde edades tempranas.

Dejando de esta forma prácticamente toda la responsabilidad de la enseñanza de lenguajes de programación a las universidades y a canales alternos (YouTube, plataformas de enseñanza en línea como UDEMY, Platzi, etc). Su consecuencia, como no podría ser de otra forma es la escasez de programadores calificados para desarrollar en ciertos lenguajes de programación obligando a

los directores de Tecnologías de la Información (TI) a focalizar los desarrollos exclusivamente en ciertos lenguajes de programación. Esta decisión generalmente se hace sin una planeación previa ni mucho menos sustentar la elección en medios teóricos ni marcos de trabajo que ayuden a tomar una decisión correcta.

Sin embargo, y a pesar de las dificultades planteadas, Honduras actualmente está iniciando un auge que le brinda mayor relevancia a las Tecnologías de la Información (TI) y poco a poco se muestra mayor interés en lo que TI puede aportar a la empresa en el cumplimiento de objetivos estratégicos.

Una prueba de ello es la encuesta realizada en el evento llamado “Tech Day Honduras 2024” en el que se muestra que un total de 33 organizaciones tienen planes de migrar su infraestructura de servidores a proveedores de servicios en la nube tales como Azure o Amazon Web Services (NOW, 2024), lo cual demuestra la aceptación y relevancia que TI que está teniendo en la actualidad en las empresas.

Esto va de la mano con el aspecto económico ya que un lenguaje de programación influye en este aspecto porque necesita de una infraestructura donde se ejecutará, su influencia es tanta que los precios de proveedores en la nube varían según el sistema operativo utilizado como servidor. Por ejemplo, en Amazon Web Services tener una instancia del modelo t3a.xlarge con sistema operativo Windows tiene un precio bajo demanda de 0,224 USD y de 0,1504 USD para Linux (Amazon Web Services, 2024).

En el caso de lenguajes como PHP, JavaScript y Python su rendimiento es equilibrado entre los distintos sistemas operativos y su versatilidad no impide que los mismos puedan ejecutarse exclusivamente en un sistema operativo.

Hablando sobre elementos sociales, Honduras no cuenta con un grupo de personas que formen una comunidad para brindar soporte a un lenguaje de programación y que esto resulte determinante para seleccionarlo, sin embargo, según mi experiencia en el mercado laboral mucho tiene que ver el rubro y el grupo objetivo de dispositivos de destino para desarrollar en cierto lenguaje de programación.

Por ejemplo, si la empresa es una manufacturera o de textiles se suele utilizar C# por su facilidad de desarrollo e integración con sistemas de terceros y su facilidad para escalar las

aplicaciones hasta un entorno web por medio del framework ASP.NET. La convergencia entre ambos lenguajes es clara, puesto que ambos son desarrollados por Microsoft y ASP cuenta con una sintaxis muy parecida a la de C# facilitando su migración y disminuyendo la curva de aprendizaje del equipo de desarrollo.

## **2.3 Teorías De Sustento**

### **2.3.1 Cuarta Revolución Industrial**

La cuarta revolución industrial, que comenzó en la primera década del siglo XXI, se caracteriza por una transformación tecnológica sin precedentes, impulsada por la convergencia de tecnologías digitales, físicas y biológicas. A diferencia de las revoluciones anteriores, esta etapa no se centra exclusivamente en la mecanización, la electrificación o la automatización, sino en la integración masiva de sistemas inteligentes y conectados que están remodelando la forma en que las sociedades interactúan, trabajan y se desarrollan (Klaus Schwab, 2016).

En el ámbito económico, la cuarta revolución industrial ha permitido la aparición de economías digitales, donde los datos se han convertido en uno de los recursos más valiosos. La automatización y la conectividad han impulsado la eficiencia y la personalización de productos y servicios, pero también han generado desafíos relacionados con la disrupción laboral, la ciberseguridad y la brecha digital.

A nivel social, esta revolución ha transformado profundamente la forma en que las personas interactúan entre sí y con la tecnología. Las redes sociales, las plataformas de comercio electrónico y las aplicaciones móviles han redefinido las dinámicas de consumo y comunicación. Al mismo tiempo, las tecnologías emergentes han abierto nuevas oportunidades en educación, salud y entretenimiento, permitiendo un acceso más equitativo y personalizado a estos servicios.

Según Schwab (2016), la tecnología y la digitalización lo revolucionaran todo, las innovaciones tecnológicas más importantes aún no han sido dadas a conocer al mundo entero y están a punto de generar un cambio trascendental aun mayor en el mundo, algo que es inevitable.

Las innovaciones tecnológicas están estrechamente ligadas a los lenguajes de programación, los cuales convierten en realidad las ideas más ambiciosas del ser humano. Gracias a ellos, lo que alguna vez fue intangible se materializa, transformándose en herramientas y soluciones que generan un impacto positivo para el beneficio de la humanidad.



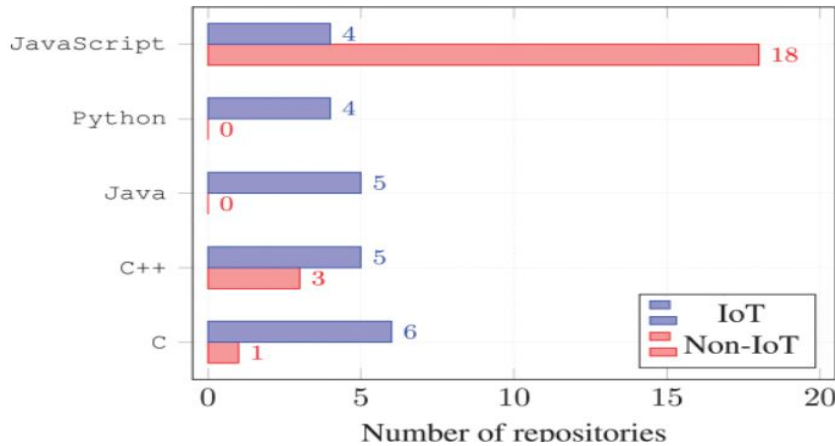
### Ilustración 2 Revoluciones Industriales

Fuente: Schwab (2016)

Esta revolución ha venido acompañada con tecnologías vanguardistas que utilizan un lenguaje de programación para funcionar:

- **Big Data:** con el auge de esta herramienta se busca analizar un gran conjunto de datos para tomar una decisión clave según sea el rubro de la empresa, o para determinar el posible motivo de comportamientos sociales. Esta herramienta está estrechamente relacionada con un lenguaje de programación puesto que ciertos lenguajes incluyen bibliotecas y librerías especiales que facilitan el procesamiento, análisis y visualización de datos para emplearlos en el Big Data (Rodríguez-Herrera et al., 2023). Hablar de Big Data comúnmente se relaciona con Python y R dejando de lados a otros lenguajes como Java, JavaScript, PHP, etc.
- **Internet de las Cosas (IoT):** este concepto emergente ha revolucionado el día a día de la humanidad, su objetivo se basa en controlar aparatos físicos por medio del software y es aquí donde los lenguajes de programación entran en escena. Con el pasar del tiempo los lenguajes han evolucionado para poder ser una opción para aplicarse en esta tecnología.

Según un estudio realizado en 2020 por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) que muestra la preferencia por el uso de C en proyectos de IoT por encima de otros lenguajes (IEEE et al., 2020).



**Ilustración 3 Lenguajes de Programación en IoT**

Fuente: IEEE (2020)

### 2.3.2 ITIL

El marco de gestión propuesto por Information Technology Infrastructure Library (ITIL) y sus prácticas puede influir significativamente en la elección de un lenguaje de programación. Entre estas prácticas, destaca "Desarrollo y Gestión de Software", que establece directrices para garantizar que las herramientas y soluciones desarrolladas se alineen con los objetivos del negocio y las necesidades del servicio.

Esta práctica define que las aplicaciones de software independientemente de si son desarrolladas internamente o por un tercero son vitales para otorgar valor al cliente en los servicios habilitados por tecnología, lo cual demuestra la importancia de esta práctica en organizaciones de TI modernas. (AXELOS, 2019).

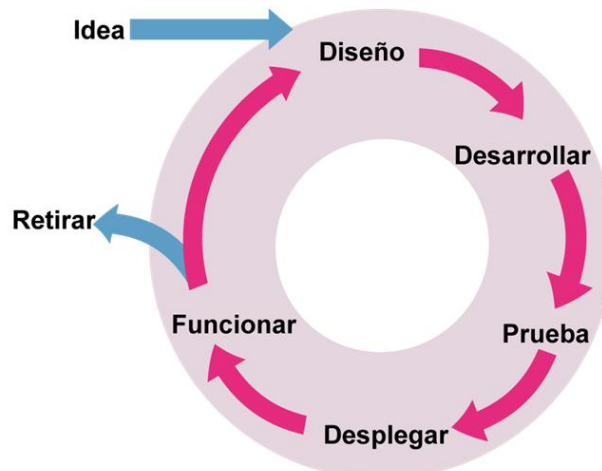
La práctica va más allá de las aplicaciones ya que abarca un conjunto de actividades que resultan claves a la hora de elegir un lenguaje de programación para desarrollar una nueva aplicación o para adquirir una por un tercero:

- **Arquitectura de soluciones:** el lenguaje de programación deber ser compatible con la arquitectura que sea elegida (monolítica, microservicios) y debe permitir construir soluciones escalables.
- **Diseño de soluciones:** deberá facilitar el diseño de las soluciones cumpliendo los requisitos previamente recabados.
- **Pruebas de software:** esto se refiere a integración entre sistemas de la empresa, la

retroalimentación del cliente y validar la funcionalidad general de la solución.

- **Control de versiones:** es un elemento para tomar en cuenta la hora de desarrollar software, tener un control de las versiones realizadas y los elementos nuevos que se incluyen y los que dejaron de estar presentes en la aplicación. Esto va de la mano de una buena documentación que permita gestionar el conocimiento del equipo de buena manera.
- **Mejora continua:** con el tiempo un lenguaje evoluciona lo que obliga a la empresa a evolucionar en sintonía para no contar con soluciones antiguas. Por otra parte, una empresa puede necesitar escalar la aplicación, por lo que un lenguaje con librerías que permita ampliar de forma sencilla puede resultar beneficioso.

Dentro de la práctica, se incluye un ciclo de vida de desarrollo de software el cual establece los pasos fundamentales que un nuevo desarrollo debe seguir para garantizar la entrega de valor al cliente como resultado final. Este ciclo de vida se estructura en fases claramente definidas, cada una con objetivos específicos que contribuyen a la calidad, eficiencia y alineación con las necesidades del negocio.



#### **Ilustración 4 Ciclo de vida del desarrollo de software según ITIL**

Fuente: ITIL (2019)

Al seguir este ciclo de vida basado en ITIL, los equipos de desarrollo no solo garantizan la entrega de un producto funcional, sino que también aseguran que el software esté alineado con las expectativas del cliente, cumpla con los estándares de calidad y contribuya de manera sostenible a los objetivos estratégicos de la organización, además, este enfoque estructurado facilita la gestión eficiente de los recursos, reduce riesgos y promueve la mejora continua.

## 2.4 Metodologías Temáticas

### 2.4.1 Marco de Gestión de Proyectos Ágil Scrum

El tema de gestión de proyectos ha tenido un auge importante en la actualidad, donde por los elementos cambiantes del mundo laboral, factores económicos y sociales se hace muy difícil gestionar un proyecto con un modelo de gestión tradicional. Estos modelos tradicionales tienen como principal característica la planeación anticipada y firme de los elementos de un proyecto dejando por fuera la posibilidad de adaptarse al cambio.

En este contexto Scrum aporta nuevos elementos que proponen un marco de trabajo diferente a lo que normalmente se acostumbra. Scrum se basa en el empirismo y el pensamiento Lean. El primer elemento hace referencia a que el conocimiento viene de la mano de la experiencia y en la toma de decisiones según lo que se observa (*Scrum Guides*, 2020).

Este marco de trabajo propone una gestión flexible que permite continuar con la ejecución del proyecto sin antes terminar una fase anterior que no sea imprescindible, de esta forma se garantiza que un proyecto no se detenga y continúe su ejecución evitando alargar el tiempo de finalización y su entrega.

De acuerdo con lo anterior, Scrum cuenta con una serie de beneficios planteados por (Hema et al., 2020):

- **Disminuir los tiempos de entrega:** dado su modelo basado en la entrega de valor de forma incremental es posible determinar posibles cuellos de botella que entorpezcan el proyecto o la solución desarrollada.
- **Comunicación constante:** propicia la comunicación en doble vía con los usuarios finales, lo cual hace al cliente parte del proyecto y permite brindar retroalimentación constante sobre el proyecto evitando entregar una solución que no cumpla con lo esperado.
- **Gestión de proyectos grandes:** en proyectos grandes se suele dividir las asignaciones por un periodo de tiempo a lo que se conoce como Sprint, un elemento que permite evaluar la aportación de un miembro al proyecto.
- **Pruebas de la aplicación al final de cada sprint:** cuando se trata de un proyecto tangible como una aplicación de software las entregas al final de cada sprint deben

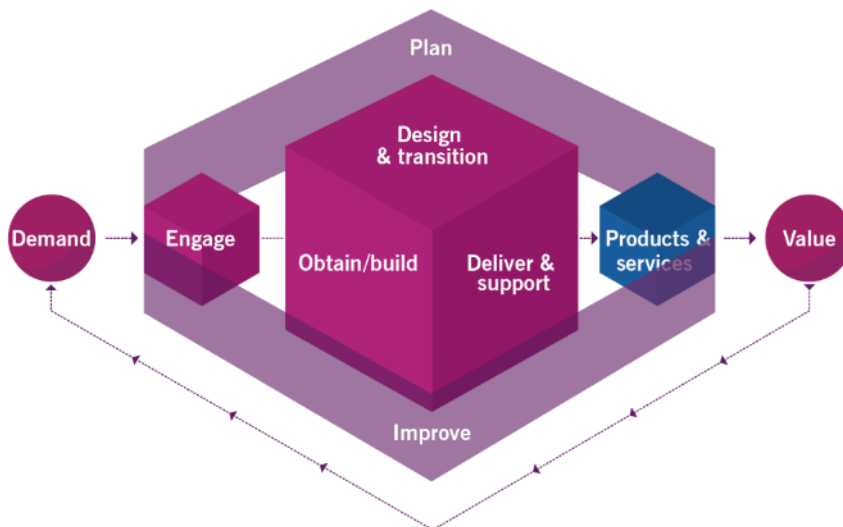
concluir con su respectiva etapa de pruebas previniendo de esta forma errores en el código y comportamientos no deseados de la aplicación.

- **Mejora continua:** dada la continua retroalimentación del cliente es posible mejorar el producto aportando valor y cumpliendo las expectativas iniciales.

### 2.4.2 Information Technology Infrastructure Library (ITIL)

ITIL se basa en un ciclo de vida del servicio compuesto por cinco etapas: estrategia, diseño, transición, operación y mejora continua. Este marco permite a las organizaciones adaptarse a cambios rápidos en las necesidades de TI, garantizando la entrega de valor al cliente y la alineación con los objetivos del negocio. Estas etapas están diseñadas para garantizar que los servicios de TI sean eficientes, alineados con las metas de negocio y capaces de adaptarse a un entorno cambiante.

Además, ITIL presenta la cadena de valor del servicio el cual representa un modelo operativo integral que describe cómo las organizaciones de TI pueden crear, entregar y mejorar servicios que aporten valor tanto a los clientes como al negocio. Este enfoque está intrínsecamente conectado al ciclo de vida del servicio en ITIL, proporcionando una perspectiva más moderna y adaptable para enfrentar las demandas cambiantes del entorno tecnológico y empresarial (AXELOS, 2019).



#### Ilustración 5 Cadena de Valor del Servicio ITIL

Fuente: ITIL (2019).

A continuación, se presenta una descripción de los elementos propuestos por ITIL con un enfoque

orientado al desarrollo de software y a la elección de un lenguaje de programación:

- **Planificación:** Evaluar las opciones disponibles en función de los requisitos del proyecto, la experiencia del equipo, la escalabilidad, la comunidad, y la disponibilidad de herramientas y librerías.
- **Diseño y Transición:** Crear prototipos funcionales para validar las decisiones de diseño y obtener retroalimentación de forma temprana que permita identificar si se cubre con las necesidades del usuario final.
- **Obtener y Construir:** Asegurar que el equipo tenga las habilidades necesarias para trabajar con el lenguaje de programación seleccionado.
- **Entrega y Soporte:** Implementar el software en el entorno de producción de manera segura y controlada y supervisar el rendimiento del software y detectar proactivamente cualquier problema.
- **Mejora Continua:** Con el tiempo y como en cualquier proyecto es esencial mejorar el producto final, es por ello por lo que se debe mejorar la calidad y mantenibilidad del código a lo largo del tiempo y esto incluye mantenerse actualizado con las últimas tendencias en desarrollo de software y lenguajes de programación que puedan adaptarse y permitir escalar la aplicación según la necesidad.

## 2.5 Herramientas e Instrumentos

### 2.5.1 Marco de Trabajo Ágil Scrum

Un proyecto sin un modelo para administrarlo no tendría rumbo y posiblemente fracase en su desarrollo. Scrum aporta muchos elementos que son vitales a la hora de gestionar proyectos grupales o individuales, entre ellos se destaca su facilidad y flexibilidad que permite adaptarse de forma rápida a elementos cambiantes del mundo laboral. Su enfoque y modelo incremental tiene similitud con un principio guía de ITIL “progresar de manera iterativa con retroalimentación”, lo cual nos indica que este es un marco que promueve la colaboración entre los integrantes.

Scrum, un marco ágil, se destaca por su enfoque iterativo e incremental, donde el trabajo se divide en ciclos cortos llamados sprints. Cada sprint culmina con una entrega funcional, permitiendo una adaptación constante a los cambios y una retroalimentación continua con el cliente. Esta flexibilidad es crucial en un entorno cada vez más dinámico y competitivo.

Este modelo cuenta con las siguientes ventajas en el ámbito de proyectos e investigativo:

**Tabla 5 Ventajas SCRUM**

Ventaja	Descripción en Gestión de Proyectos	Descripción en Investigación
<b>Adaptabilidad y flexibilidad</b>	Scrum permite responder rápidamente a cambios en los requisitos o en el entorno del proyecto gracias a los sprints cortos y revisiones constantes.	Permite ajustar el enfoque de la investigación en función de hallazgos preliminares o cambios en las hipótesis, mejorando la relevancia de los resultados.
<b>Entrega incremental de valor</b>	El producto se desarrolla en incrementos, entregando valor de forma continua y permitiendo obtener retroalimentación del cliente rápidamente.	Fomenta la obtención de resultados parciales a lo largo del proceso, facilitando el análisis preliminar y ajustando la dirección de la investigación.
<b>Mejora continua</b>	A través de retrospectivas en cada sprint, el equipo identifica áreas de mejora, optimizando procesos y eficiencia constantemente.	Las retrospectivas permiten a los investigadores ajustar métodos y técnicas, mejorando el rigor y la efectividad de la investigación progresivamente.
<b>Reducción de riesgos</b>	Los sprints cortos y la revisión continua permiten identificar y mitigar riesgos antes de que escalen.	Permite identificar y corregir posibles errores metodológicos o de enfoque en fases tempranas, minimizando el riesgo de resultados inexactos o irrelevantes.
<b>Mayor transparencia</b>	Las reuniones diarias y conceptos de Scrum como el backlog proporcionan visibilidad sobre el estado del proyecto a todas las partes.	La transparencia ayuda a los equipos de investigación a alinear objetivos y mantener claridad sobre el progreso y las dificultades, fomentando la colaboración.
<b>Enfoque en el cliente o usuario</b>	La participación constante del cliente asegura que el producto final cumpla con las expectativas y requisitos del usuario.	Facilita la orientación de la investigación hacia las necesidades de la comunidad o de los usuarios finales.
<b>Mejor comunicación y colaboración</b>	Scrum fomenta la comunicación efectiva entre el equipo mediante reuniones diarias y roles bien definidos, reduciendo malentendidos y errores.	La comunicación constante ayuda a mantener la cohesión entre los investigadores, mejorando la integración de ideas y datos entre disciplinas o áreas de estudio.

Ventaja	Descripción en Gestión de Proyectos	Descripción en Investigación
<b>Priorización efectiva de tareas</b>	El backlog del producto permite priorizar tareas según su impacto, facilitando un enfoque en las funcionalidades de mayor valor.	En investigación, permite concentrarse en los objetivos o hipótesis más importantes, optimizando recursos y tiempo hacia los aspectos más relevantes.
<b>Empoderamiento del equipo</b>	Los equipos Scrum tienen autonomía para organizar su trabajo, lo que mejora la motivación y la responsabilidad en la entrega de resultados.	En investigación, el equipo tiene la libertad de decidir cómo abordar problemas o hipótesis, lo que fomenta la creatividad y el pensamiento crítico.
<b>Retroalimentación continua</b>	Las revisiones de sprint permiten al equipo recibir feedback constante, asegurando que el proyecto esté alineado con las expectativas.	La retroalimentación continua permite ajustar el diseño experimental o los enfoques metodológicos, mejorando la calidad y relevancia de la investigación.
<b>Rapidez en la detección de problemas</b>	Las reuniones diarias y la revisión constante facilitan la detección y resolución rápida de problemas.	Identificar problemas de método o recolección de datos en etapas tempranas reduce el impacto negativo en los resultados finales de la investigación.
<b>Facilita la innovación</b>	La estructura iterativa y el enfoque en la mejora continua proporcionan un ambiente favorable para la innovación en cada sprint.	Los ajustes y experimentos en cada iteración permiten probar enfoques innovadores y explorar nuevas hipótesis en un entorno controlado.

Fuente: Elaboración propia

A lo largo de los años, se han desarrollado diversas metodologías de gestión de proyectos, cada una con características específicas diseñadas para abordar diferentes tipos de desafíos. En el ámbito del desarrollo de software y la gestión ágil de proyectos, destacan tres enfoques principales: Scrum, Kanban y Extreme Programming (XP).

Scrum ha ganado una enorme popularidad en el mundo del desarrollo de software debido a su enfoque estructurado, iterativo e incremental, que permite gestionar la complejidad de los proyectos de manera eficaz. Sin embargo, Kanban y Extreme Programming (XP) también ofrecen ventajas significativas y pueden ser adecuados en determinados escenarios. Para comprender mejor las diferencias y justificar la elección de Scrum, a continuación, se presenta una tabla comparativa de estas metodologías:

**Tabla 6 Comparativa de Scrum con otras herramientas**

Aspecto	Scrum	Kanban	Extreme Programming (XP)	Ventaja de Scrum
<b>Estructura</b>	Basado en sprints, con iteraciones fijas (1-4 semanas).	Flujo continuo de trabajo sin iteraciones definidas.	Iteraciones cortas (1-2 semanas).	Ideal para proyectos que requieren entregas periódicas.
<b>Roles definidos</b>	Scrum Master, Product Owner, y Equipo de desarrollo.	Sin roles formales específicos.	Cliente, Programadores, Tester.	Promueve la claridad en responsabilidades.
<b>Planificación</b>	Sprint Planning al inicio de cada iteración.	Las tareas se agregan al tablero en cualquier momento.	Planificación centrada en historias de usuario.	Mejor para planificar entregas y prioridades.
<b>Entrega de valor</b>	Entregas al final de cada sprint.	Entregas continuas cuando una tarea se completa.	Entregas pequeñas y frecuentes de valor.	Favorece la alineación con fechas críticas.
<b>Monitoreo de progreso</b>	Se mide con gráficos.	Se usa el diagrama de flujo acumulado	Gráficos de velocidad y métricas de calidad.	Mejora la visibilidad de progreso por sprint.
<b>Cambio en el trabajo</b>	No se permiten cambios en el sprint en curso.	Cambios permitidos en cualquier momento.	Cambios posibles en iteraciones futuras.	Ayuda a mantener el enfoque del equipo.
<b>Reuniones</b>	Incluye reuniones específicas: Daily Scrum, Sprint Review, Sprint Retrospective.	Sólo reuniones según necesidad.	Reuniones diarias y revisiones frecuentes.	Estructura las interacciones para mejorar procesos.
<b>Mejora continua</b>	Se enfoca en retrospectivas al final de cada sprint.	Mejora continua basada en métricas del flujo.	Refactorización continua y revisiones técnicas.	Estructura formal para aprendizaje iterativo.
<b>Uso típico</b>	Proyectos con requisitos cambiantes y objetivos claros.	Operaciones continuas o equipos de soporte.	Proyectos de software donde la calidad técnica es prioritaria.	Más adecuado para desarrollos iterativos y por fases.

Fuente: Elaboración propia.

## 2.5.2 Herramientas Tecnológicas

El uso de herramientas tecnológicas en la investigación no solo optimiza el tiempo y los recursos, sino que también potencia la calidad de los resultados obtenidos. Estas herramientas permiten agilizar procesos, incrementar la eficiencia en la ejecución de tareas, mejorar la precisión en la recolección de datos, y facilitar el análisis y presentación de la información de manera más profesional y comprensible.

En este contexto, las tecnologías no solo se limitan a la recopilación de datos, sino que también ofrecen ventajas significativas en la organización, almacenamiento y manejo de grandes volúmenes de información. Además, permiten realizar simulaciones, análisis estadísticos complejos y visualizaciones dinámicas que enriquecen la comprensión de los hallazgos.

En la investigación propuesta se emplearán diversas herramientas que contribuirán al desarrollo del proyecto en distintas facetas de la investigación aportando valor en el desarrollo de esta.

Las herramientas que se utilizarán durante el desarrollo del proyecto investigativo son:

**Tabla 7 Comparativa de Herramientas**

Categoría	Herramienta	Alternativas Similares	Ventajas de la herramienta mencionada
<b>Computadora</b>	Computadora estándar	Tablets, Chromebooks	Mayor potencia de procesamiento, versatilidad para software avanzado, opciones de personalización y capacidad de ampliación
<b>Software de gestión de proyectos</b>	Trello	Asana, Monday.com, Jira	Interfaz intuitiva, simplicidad para pequeños equipos, integración con múltiples herramientas y un modelo freemium accesible
<b>Software para la recolección de datos</b>	Microsoft Forms	SurveyMonkey, Google Forms	Fácil de usar, integración con otras herramientas (como Excel), opciones personalizables y herramientas personalizadas para análisis de datos
<b>Software para el análisis de datos</b>	Power Bi, Excel, Knime, Python	Qlik, Tableau	Visualizaciones avanzadas, soporte amplio de datos, integración con diferentes fuentes y facilidad de uso para todo tipo de usuarios.

Fuente: Elaboración propia

### **2.5.3 Instrumentos de recolección de datos**

El trabajo investigativo se apoyará de una encuesta exhaustiva dirigida a expertos en desarrollo de software con una experiencia consolidada en la selección y el uso de lenguajes de programación. La elección de una entrevista permite obtener una combinación de datos cuantitativos y cualitativos, necesarios para abordar de manera completa y representativa el tema de investigación sobre la selección de lenguajes de programación y su relación con prácticas de gestión como ITIL.

El diseño de la encuesta incluirá preguntas estructuradas, semiestructuradas y abiertas, lo que permitirá no solo cuantificar tendencias y patrones en las respuestas, sino también explorar percepciones y experiencias únicas de los participantes (Hernández Sampieri et al., 2010). Este enfoque mixto asegura la obtención de datos representativos y de alta calidad, esenciales para extraer conclusiones sólidas y formular recomendaciones aplicables tanto a nivel técnico como estratégico en el ámbito de la gestión tecnológica y el desarrollo de software. La combinación de esta herramienta con un análisis detallado proporcionará un panorama claro y fundamentado sobre cómo las decisiones en la selección de lenguajes de programación impactan en la implementación y el alineamiento con marcos como ITIL, contribuyendo así al avance del conocimiento en este campo.

La elección de estos participantes responde a su conocimiento avanzado y su habilidad para comprender y evaluar de manera profunda los factores críticos involucrados en la elección de un lenguaje de programación. Estos expertos cuentan con varios años de experiencia en la industria, lo que asegura no solo la competencia técnica, sino también la exposición repetida a escenarios complejos en los que la selección de lenguajes juega un papel determinante en el éxito del proyecto.

## **2.6 Conceptualización**

### **Algoritmo**

Describe una serie ordenada de pasos o instrucciones diseñadas para resolver un problema o realizar una tarea específica de manera eficiente y lógica, de modo que puedan ser interpretadas y ejecutadas por un ordenador. Un aspecto interesante es que, para un mismo problema, a menudo existen múltiples algoritmos posibles que pueden ser propuestos como soluciones. Cada uno de estos métodos puede diferir en su enfoque, eficiencia, complejidad (Urquía Moraleda, 2021).

## **Lenguaje de Programación**

Es una herramienta compuesta por reglas, sintaxis y semántica que permite escribir instrucciones para que las computadoras puedan ejecutarlas. Es el medio de comunicación entre los seres humanos y las máquinas, diseñado para traducir ideas o algoritmos en un formato que las computadoras puedan interpretar y procesar (Universidad ORT Uruguay, 2024).

## **Inteligencia Artificial**

Se trata de un conjunto de tecnologías innovadoras diseñadas para que las computadoras realicen tareas avanzadas que antes requerían inteligencia humana. Estas capacidades incluyen la interpretación, comprensión y traducción del lenguaje hablado y escrito, así como el análisis de grandes volúmenes de datos, la identificación de patrones, la formulación de recomendaciones personalizadas y mucho más (Google Cloud, 2024).

## **Software**

El software es la parte intangible de un sistema informático, se le conoce como el producto final de un proyecto de programación que interactúa y hace de intermediario entre las acciones ordenadas por el humano mediante el hardware y la computadora.

## **Hardware**

Son componentes físicos del sistema informático que permiten al usuario manipular un software mediante operaciones mecánicas que requieren la acción de un humano.

## **Framework**

Herramienta construida basada en un lenguaje de programación base, su diferencia radica en que proponen una nueva forma de construir aplicaciones con los nuevos elementos incluidos, esto garantiza la calidad del software final ya que su base ha sido reconstruida y probada de forma exhaustiva (Velásquez-Restrepo et al., 2018).

## **Information Technology Infrastructure Library (ITIL)**

Es un marco de trabajo diseñado para estandarizar, gestionar y optimizar los servicios de TI dentro de una organización. Proporciona un conjunto de mejores prácticas que ayudan a las empresas a gestionar los servicios de TI para aportar valor a las empresas (IBM, 2021).

## **Infraestructura**

La infraestructura tecnológica es definida como un conjunto de elementos integrados por hardware (servidores, computadora, laptop, etc.), software (internet, lenguajes de programación,

etc.). también, la infraestructura tecnológica se refiere a todo aquello que está relacionado, con las tecnologías de la información y comunicaciones (Hinojosa Mamani et al., 2023).

### **Servidor**

Encargado de responder las solicitudes de otros programas. Actúa como un intermediario que gestiona y proporciona recursos, servicios o datos solicitados por los clientes, facilitando la comunicación y el intercambio de información.

### **Arquitectura**

La arquitectura de software de un sistema se define como el conjunto organizado de estructuras fundamentales que forman la base para entender, diseñar y gestionar dicho sistema. Estas estructuras incluyen no solo los componentes principales de software, sino también la forma en que estos interactúan entre sí, las conexiones que establecen, y las propiedades o atributos clave que caracterizan tanto a los elementos individuales como a sus relaciones (Cervantes, 2016).

### **Microservicio**

Basado en un conjunto de servicios autónomos que pueden desarrollarse y gestionarse de manera independiente. Cada servicio cuenta con su propia lógica empresarial y base de datos, diseñados para cumplir un propósito específico. Las actualizaciones, pruebas, implementaciones y escalabilidad se realizan de forma aislada dentro de cada uno de estos servicios (Atlassian, 2024).

### **Arquitectura Monolítica**

Es una estructura informática unificada y amplia, donde todo el código fuente integra las funcionalidades empresariales en un único sistema. Para realizar modificaciones en este tipo de aplicación, es necesario actualizar toda la pila tecnológica, lo que implica acceder al código fuente, compilarlo y desplegar una nueva versión de la interfaz del lado del servicio. Este enfoque dificulta las actualizaciones, haciéndolas más rígidas y lentas (Atlassian, 2024).

## **2.7 Marco Legal**

### **2.7.1 Marco Legal Internacional**

Aunque no existen leyes específicas que regulen directamente qué lenguajes deben usarse, diversas normativas legales y estándares internacionales desempeñan un papel determinante en la selección del lenguaje más adecuado. Estas normativas están intrínsecamente vinculadas a aspectos críticos como la protección de datos, la seguridad de la información, y la conformidad

con requisitos técnicos específicos. Las leyes o normativas pueden ser vistas en la tabla 6.

**Tabla 8 Leyes o Normativas Internacionales**

Ley o Normativa	Descripción de su influencia en el desarrollo de software
<b>ITIL</b>	Es un conjunto de mejores prácticas ampliamente adoptado en la gestión de servicios de TI. Su influencia en el desarrollo de software se centra en garantizar que las herramientas de TI estén apegadas con los procesos de negocio, mejoren la eficiencia y satisfagan las necesidades del usuario final.
<b>COBIT (Control Objectives for Information and related technology)</b>	COBIT es un marco de trabajo para la gobernanza y gestión de TI. Influye en el desarrollo de software al establecer controles y objetivos claros para asegurar la integridad, confidencialidad y disponibilidad de la información. Las herramientas de software deben cumplir con estos controles para garantizar la seguridad y eficacia de los procesos de negocio.
<b>ISO 27001</b>	Esta norma internacional establece conjunto de requisitos para implementar y mantener un Sistema de Gestión de Seguridad de la Información (SGSI). Las herramientas de software deben cumplir con los controles especificados en la norma para proteger la información confidencial de la organización y sus clientes.
<b>GDPR (Reglamento General de Protección de Datos)</b>	GDPR es una regulación de la Unión Europea (UE) que establece un marco común para la protección de datos personales. Las herramientas de software que procesan datos personales de ciudadanos de la UE deben cumplir con los principios del GDPR, como el consentimiento, la transparencia y la seguridad de los datos.
<b>Ley de California de Protección de Datos (CCPA)</b>	Similar al GDPR, la CCPA es una ley estadounidense que otorga a los usuarios de California ciertos derechos sobre sus datos personales. Las empresas que operan en California y recopilan datos personales deben cumplir con los requisitos de la CCPA, 10 que puede influir en el desarrollo de herramientas de software diseñadas para gestionar datos

Fuente: Elaboración propia

### 2.7.2 Marco Legal Nacional

A pesar de la notable brecha que separa a Honduras de otros países en cuanto a reglamentos internacionales sobre la seguridad de la información, existen disposiciones legales que abordan la protección de datos personales. Estas normativas destacan la importancia de garantizar el cumplimiento legal en el desarrollo y uso de software, ya que la infracción de dichas leyes puede generar conflictos legales significativos para las empresas.

El incumplimiento de estas disposiciones, como las relacionadas con el consentimiento del usuario, la protección de datos sensibles o la transferencia de información personal a terceros sin garantías de seguridad adecuadas, puede exponer a las organizaciones a sanciones legales y demandas por parte de los titulares de los datos. Esto recalca la necesidad de desarrollar soluciones tecnológicas que integren medidas de protección de datos desde su diseño y aseguren el cumplimiento normativo en todas las etapas de su operación.

**Tabla 9 Leyes o Normativas Nacionales**

Ley o Normativa	Descripción
<b>Decreto 130-2017</b>	En el artículo 272 se dicta castigo a quienes vulneren la identidad y los datos personales, enfatizando la necesidad de medidas estrictas para prevenir accesos no autorizados o abusos de información personal
<b>Normativa para regular la administración de Tecnología en instituciones financieras por la Comisión Nacional de Banca y Seguros (CNBS)</b>	Exponen un conjunto de requisitos que se deben cumplir en el área de TI para que una empresa pueda operar en el rubro financiero. Su objetivo es garantizar la seguridad de los datos y mantener una infraestructura tecnológica capaz de satisfacer las necesidades.
<b>PCM-089-2020</b>	En varias secciones del artículo se hace referencia a la protección de datos personales y que todas las personas tienen este derecho. En el documento se insta a las empresas a implementar medidas para cumplir este objetivo y resguardar la privacidad de los usuarios.

Fuente: Elaboración propia

Todas las regulaciones cuentan con el objetivo de salvaguardar y proteger los datos de los clientes. Estas normativas implican desafíos técnicos y su influencia directa con la elección de un lenguaje de programación radica en que los proyectos de software deben cumplir con estos requerimientos desde la etapa de planificación.

## CAPÍTULO III – METODOLOGIA DE LA INVESTIGACION

Los capítulos anteriores han destacado lo complicado que puede ser elegir el lenguaje de programación adecuado para el desarrollo web. Además, muestran cómo una correcta gestión y uso de marcos de trabajo como ITIL, SCRUM y algunos principios de COBIT pueden beneficiar a las empresas, ayudándolas a ser más eficientes y a obtener mejores resultados.

El objetivo principal de esta investigación es comprender la importancia de la elección adecuada de un lenguaje de programación en el contexto de proyectos de desarrollo de software, con énfasis en aplicaciones web y determinar el efecto que esto puede tener en una empresa si se elige de forma adecuada o equivocada.

Para lograr este objetivo, se utilizará un enfoque basado en la recolección y análisis de datos primarios. La principal técnica será la aplicación de encuestas dirigidas a expertos del área del desarrollo web, incluidos desarrolladores, arquitectos de software, y líderes de proyectos con experiencia en el uso de diferentes lenguajes y marcos de trabajo.

### 3.1 Tipo de enfoque

El enfoque de la investigación es mixto, ya que la herramienta de recolección de datos permite obtener diferentes tipos de datos que otorgan la facilidad de analizar datos cualitativos y cuantitativos por medio de análisis estadísticos que permitirán ofrecer resultados precisos.

La elección de un enfoque mixto (con prevalencia cualitativa) en esta investigación responde a la necesidad de examinar el impacto de la selección de lenguajes de programación desde diversas perspectivas. Por un lado, los datos cuantitativos, obtenidos mediante encuestas estructuradas, permitirán realizar análisis estadísticos que revelen patrones y tendencias. Por otro lado, los datos cualitativos, recopilados a través de preguntas abiertas, proporcionarán una comprensión más profunda de las percepciones, experiencias y contextos que influyen en las decisiones de los expertos en desarrollo web.

La fortaleza del enfoque mixto radica en su capacidad para responder a un problema de investigación utilizando un diseño metodológico que puede ser (Otero, 2018):

- **Concurrente:** La recolección de datos se efectúa por medio de una encuesta en línea, lo que indica que la captación de todos los tipos de datos se realiza de forma simultánea. Esto permite comparar y contrastar las respuestas obtenidas además de evitar sesgos temporales

ya que evita realizar recolecciones escalonadas.

- **De conversión:** Los datos cualitativos pueden cuantificarse, lo que facilita el análisis desde perspectivas diferentes y enriquecedoras.
- **De integración:** Combina completamente ambos tipos de datos en cada etapa del proceso de investigación, desde el diseño hasta el análisis, para producir resultados que sean más representativos y holísticos.

### **3.2 Alcance de la investigación**

Con los estudios descriptivos se busca especificar las propiedades, las características y los perfiles de personas, grupos, comunidades, procesos, objetos o cualquier otro fenómeno que se someta a un análisis (Hernández Sampieri et al., 2010).

El alcance de la presente investigación cuenta con un modelo descriptivo puesto que se intentará realizar un diagnóstico de como un equipo de desarrollo elige un lenguaje de programación y la influencia que esto puede tener. Por medio de este enfoque se intentará demostrar las características principales de los equipos de desarrollo y se podrá determinar el punto de similitud que hace que elijan un lenguaje adecuado o equivocado.

El alcance descriptivo permite caracterizar el estado actual del problema, proporcionando una visión clara de las prácticas, decisiones y criterios utilizados en la industria del desarrollo web.

Esto incluye aspectos técnicos, como el rendimiento y la escalabilidad de los lenguajes, y factores contextuales, como la experiencia del equipo, los objetivos del proyecto y las limitaciones presupuestarias o de tiempo. Este análisis no solo permite comprender el panorama actual, sino que también puede actuar como una base para explorar áreas de mejora, identificar vacíos en el conocimiento y generar nuevas oportunidades para optimizar la elección tecnológica.

Un componente clave de esta investigación será la elaboración de una guía práctica para la selección de lenguajes de programación, destinada a ayudar a equipos y organizaciones a tomar decisiones más informadas y alineadas con sus objetivos. Esta guía se desarrollará a partir de los hallazgos obtenidos en el análisis descriptivo y tendrá como objetivo proporcionar criterios claros y aplicables.

### **3.3 Diseño de la investigación**

En esencia, el diseño de la investigación actúa como una hoja de ruta que organiza y dirige

las acciones necesarias para alcanzar los resultados esperados. Al proporcionar una estructura clara y estratégica, el diseño no solo facilita la organización del estudio, sino que también garantiza que los métodos utilizados sean apropiados y efectivos para responder a las preguntas de investigación planteadas. Esto es fundamental para mantener la validez de las conclusiones, asegurando que los resultados sean representativos y relevantes (Hernández Sampieri et al., 2010).

El diseño no experimental es una elección adecuada para el desarrollo de esta investigación, ya que el objetivo principal es observar, analizar y describir las dinámicas relacionadas con la elección de lenguajes de programación en equipos de desarrollo y su impacto en las organizaciones, sin intervenir directamente en el fenómeno estudiado. Este enfoque es consistente con la naturaleza descriptiva de la investigación, que busca caracterizar un problema existente en su contexto natural y recopilar información a partir de datos reales y experiencias de los participantes (Hernández Sampieri et al., 2010).

### **3.3.1 Población**

En esta investigación, la población no tiene un tamaño definido, por lo que podría considerarse infinita. Sin embargo, dado el enfoque profesional de este estudio en el ámbito del desarrollo de software, se define la población como el conjunto de todos los profesionales relacionados con este rubro. Para facilitar el análisis, se propone dividir la muestra en grupos específicos según características relevantes para el estudio.

### **3.3.2 Muestra**

La presente investigación empleará una técnica de muestreo probabilístico, específicamente un muestreo estratificado. Los estratos o grupos se dividen en desarrolladores senior, desarrolladores de software senior nivel medio y desarrolladores de software junior y líderes de proyectos, quienes han estado involucrados activamente en procesos críticos relacionados con la selección de lenguajes de programación. Estas decisiones, que frecuentemente tienen implicaciones técnicas y estratégicas, forman parte de sus responsabilidades habituales, lo que los convierte en una fuente valiosa de conocimiento y perspectivas prácticas para esta investigación. La selección de esta población responde a la necesidad de obtener información precisa y relevante basada en experiencias reales.

### **3.3.3 Técnicas de muestreo**

El uso de esta técnica asegura que cada subgrupo o estrato de la población esté

adecuadamente representado, también permite una selección sistemática y organizada de participantes, lo que facilita la comparación entre grupos y mejora la precisión de los resultados. Este enfoque garantiza que la muestra refleje la diversidad de características y perspectivas presentes en la población.

- **Muestreo estratificado:** en esta es una técnica de muestreo probabilístico en la que la población se divide en subgrupos homogéneos llamados estratos. Estos estratos se forman en función de características específicas que son relevantes para la investigación y es especialmente útil cuando se quiere comparar resultados entre grupos o segmentos, lo cual se adapta perfectamente a la presente investigación (Hernández Sampieri et al., 2010).

El cálculo del tamaño de la muestra se define por la siguiente fórmula:  $n = \frac{Z^2 PQ}{E^2}$

**Tabla 10 Tamaño de la Muestra**

Variable	Descripción	Valor
<b>Z</b>	Valor crítico correspondiente a un coeficiente de confianza del cual se desea hacer la investigación	1.645 representando un valor de confianza del 90%
<b>P</b>	Es la proporción estimada de la población que tiene la característica de interés.	0.5 al no conocer el valor exacto y ser una población infinita
<b>Q</b>	Es la proporción complementaria	1 - P
<b>E</b>	Error Muestral	0.1, se evaluará bajo un porcentaje de error del 10%
<b>n</b>	Tamaño final de la muestra luego de aplicar la fórmula	El total de la muestra es de 68.

Fuente: Elaboración propia

### 3.4 Criterios de selección de la muestra

En la presente investigación, se han establecido criterios específicos para garantizar que los datos recopilados provengan de participantes que cumplan con ciertas características clave, asegurando así que su contribución sea relevante y valiosa para los objetivos del estudio. Estos criterios actúan como filtros para seleccionar a los participantes más adecuados, lo que permite obtener información de alta calidad y con un alto grado de aplicabilidad en el contexto de la investigación. Los detalles pueden ser vistos en la siguiente tabla.

**Tabla 11 Criterios de Inclusión y Exclusión**

Categoría	Criterios de Inclusión	Criterios de Exclusión
<b>Experiencia laboral</b>	Contar con un mínimo de 2 años de experiencia en el desarrollo de software.	Tener menos de 2 años de experiencia en el desarrollo de software.
<b>Área de especialización</b>	Tener experiencia comprobada en desarrollo web y en la selección de lenguajes de programación.	No contar con experiencia en desarrollo web o en la toma de decisiones tecnológicas.
<b>Rol profesional</b>	Ocupación actual o previa como desarrollador de software, arquitecto de software o líder técnico.	No haber desempeñado un rol técnico o estratégico en proyectos de desarrollo de software.
<b>Disponibilidad</b>	Disposición para participar en encuesta relacionada al estudio.	Falta de disponibilidad o negativa para completar el proceso de recolección de datos.

Fuente: Elaboración propia

### 3.5 Hipótesis

En la presente investigación no se ha formulado una hipótesis, una decisión que se sustenta en la naturaleza del estudio y en la coherencia con su enfoque metodológico. Aunque las hipótesis son herramientas fundamentales en ciertas investigaciones, su formulación no es obligatoria en todos los casos, especialmente en aquellos estudios cuyo propósito no es comprobar relaciones causales o predictivas, sino explorar y describir fenómenos. (Hernández Sampieri et al, 2014).

La decisión de no incluir una hipótesis se fundamenta principalmente en el alcance descriptivo y el diseño no experimental de esta investigación. El alcance descriptivo se orienta a caracterizar y describir un fenómeno tal como ocurre en su contexto natural, en este caso, la selección de lenguajes de programación en equipos de desarrollo y su impacto en las organizaciones. Este alcance no tiene como objetivo identificar o validar relaciones causales ni correlacionales, sino proporcionar un diagnóstico detallado y una comprensión integral de las dinámicas implicadas (Hernández Sampieri et al., 2014).

El diseño no experimental refuerza esta postura, ya que no contempla la manipulación de variables, en consecuencia, no se pretende establecer relaciones causa-efecto. Este enfoque es congruente con el propósito de la investigación, recogiendo datos que reflejen las experiencias y percepciones de los participantes (Hernández Sampieri et al., 2014).

Aunque la investigación adopta un enfoque mixto, que integra datos cuantitativos y cualitativos, el planteamiento de una hipótesis no es un requisito indispensable en este contexto,

además, al no tratarse de un estudio con alcance correlacional o explicativo, no es necesario establecer hipótesis que busquen probar relaciones entre variables o identificar factores causales (Hernández Sampieri et al., 2014).

### 3.6 Operacionalización de las variables

A continuación, se presenta la matriz con la información correspondiente a las variables propuestas para llevar a cabo la investigación. Esta herramienta es fundamental para estructurar y organizar de manera clara y precisa los elementos clave del estudio, facilitando el proceso de análisis y asegurando la coherencia metodológica.

La matriz incluye las variables identificadas como esenciales, sus respectivas definiciones conceptuales, operativas, dimensiones y los indicadores específicos que se utilizarán para medirlas.

La información sobre las variables seleccionadas puede ser vista en la tabla 12.

**Tabla 12 Operacionalización de variables**

Variable	Tipo	Definición conceptual	Definición operacional	Indicador	Dimensión	Escala de medición
Rol Profesional	Independiente	Perfil profesional de la persona	Área a la que se dedica la persona	Número y cantidad de roles	Rol profesional	Discreta nominal
Años de experiencia profesional	Dependiente	Experiencia profesional de la persona	Experiencia general de la persona	Rango de años	Rol profesional	Intervalo
Años de experiencia en el desarrollo web	Dependiente	Experiencia en el área del desarrollo web de la persona	Experiencia específica de la persona en el rubro	Rango de años	Rol profesional	Intervalo
Metodologías en cascada	Dependiente	Utilización de metodologías tradicionales para la gestión de proyectos	Porcentaje de proyectos en los que se utiliza metodologías en cascada	Tasa de uso de herramientas para la gestión de proyectos	Herramientas de gestión de proyectos utilizada	Razón

Variable	Tipo	Definición conceptual	Definición operacional	Indicador	Dimensión	Escala de medición
Metodologías ágiles	Dependiente	Utilización de metodologías ágiles para la gestión de proyectos	Porcentaje de proyectos en los que se utiliza metodologías ágiles.	Tasa de uso de herramientas para la gestión de proyectos	Herramientas de gestión de proyectos utilizada	Razón
Conocimiento de ITIL	Independiente	Conocimiento de marco de trabajo ITIL	Escala de conocimiento que tiene de ITIL un profesional	Escala de conocimiento de ITIL	Marco ITIL	Razón
Factores relevantes (Sostenibilidad, mantenibilidad, recursos del lenguaje)	Dependiente	Aspectos que una profesional toma en cuenta, como la sostenibilidad, facilidad de aprendizaje, mantenibilidad y recursos del lenguaje	Escala de aspectos influyentes desde el punto de vista profesional	Escala de factores relevantes	Factores técnicos	Ordinal Discreta
Escala de importancia de factores propuestos (Integración, costos, curva de aprendizaje)	Dependiente	Relevancia de factores influyentes	Escala de importancia según profesionales	Rango de importancia	Factores técnicos	Ordinal Discreta
Consideración de elementos como sostenibilidad y mantenibilidad	Dependiente	Relevancia de estos factores influyentes pensando en el futuro del proyecto	Porcentaje de relevancia de factores influyentes pensando en el futuro	Porcentaje de relevancia de evaluación de aspectos del futuro	Factores técnicos	Discreta
Propósito y objetivo del proyecto	Dependiente	Identificar si los requisitos iniciales tienen influencia en la elección	Escala de importancia del propósito del proyecto	Escala de evaluación de influencia del propósito específico	Factores técnicos	Discreta

Variable	Tipo	Definición conceptual	Definición operacional	Indicador	Dimensión	Escala de medición
Experiencia del equipo de desarrollo	Dependiente	Identificar si la experiencia es un factor que suele considerarse	Porcentaje de importancia del conocimiento tecnológico del equipo	Porcentaje de influencia del stack tecnológico	Factores técnicos	Discreta
Paquetes y librerías	Dependiente	Identificar si las herramientas externas influyen	Porcentaje de relevancia de paquetes externos para el desarrollo	Porcentaje de profesionales que consideran las herramientas y paquetes	Factores técnicos	Discreta
Rubro de la empresa	Dependiente	Verificación consideración del rubro para elegir un lenguaje	Porcentaje de profesionales que consideran influencia del rubro	Porcentaje de la influencia del rubro	Factores empresariales	Discreta
Factores técnicos importantes del lenguaje (paradigmas, patrones, etc)	Dependiente	Evaluar los elementos propios del lenguaje considerados más importantes	Escala de factores técnicos considerados por los profesionales	Escala de los factores elegidos	Factores técnicos	Ordinal
Tasa de implementación de ITIL en proyectos y su influencia al elegir lenguajes	Dependiente	Identificar si ITIL es utilizado en el desarrollo de software	Porcentaje de implementación de ITIL en proyectos	Porcentaje de implementación de ITIL	Marco ITIL	Discreta
Fase del ciclo de vida de ITIL con mayor relevancia en la elección del lenguaje	Dependiente	Identificar la fase del ciclo de vida de ITIL con mayor relevancia	Porcentaje de la fase preferida del ciclo de vida de ITIL	Porcentaje de la fase mayormente seleccionada	Marco ITIL	Nominal
Principios y prácticas de ITIL	Dependiente	Identificación de la importancia y aplicación de ITIL en la empresa	Escala de uso de los principios y prácticas que tiene ITIL	Escala de uso de los principios y prácticas que tiene ITIL	Marco ITIL	Discreta

Variable	Tipo	Definición conceptual	Definición operacional	Indicador	Dimensión	Escala de medición
Experiencia donde la elección del lenguaje fue clave	Dependiente	Evaluación de experiencia	Evaluación de experiencia profesional	Identificación de panoramas a considerar	Experiencia profesional	Cualitativa
Participación en proyectos que se han tornado complejos por la equivocada elección de un lenguaje de programación	Dependiente	Conocer experiencia	Análisis de experiencias	Número de participantes con esta experiencia	Experiencia profesional	Nominal
Gestión de integrar múltiples lenguajes	Dependiente	Identificar experiencia profesional	Análisis de las experiencias	Análisis de la opinión de profesionales	Experiencia profesional	Cualitativa
Consideración de la infraestructura tecnológica	Dependiente	Identificar la infraestructura como factor	Evaluar si la infraestructura es considerable como factor clave	Número de participantes que consideran determinante la infraestructura	Experiencia profesional	Nominal
Aceptación de rúbrica	Independiente	Identificar si los profesionales tomarán en cuenta una rúbrica para ayudar en la elección	Medir el nivel de aceptación de una rúbrica para implementar	Tasa de aceptación	Rúbrica	Nominal
Factores para considerar para la rúbrica	Dependiente	Identificar los factores a tomar en cuenta	Medir los factores con preferencia	Escala de factores de rúbrica	Rúbrica	Ordinal

Fuente: Elaboración propia

### 3.7 Técnicas, instrumentos, procedimiento y plan de análisis

En esta sección se describen de manera detallada los elementos esenciales para la recolección de datos, los cuales son fundamentales para garantizar la calidad y validez de la información obtenida. Dichos elementos desempeñarán un papel crucial en el cumplimiento de los objetivos planteados en este estudio, ya que permiten recopilar, organizar y analizar los datos de manera sistemática.

La correcta implementación de estos elementos no solo facilitará el logro de los objetivos, sino que también contribuirá a que los resultados obtenidos sean representativos y confiables, fortaleciendo la base del análisis y las conclusiones del estudio.

### **3.7.1 Técnicas**

Para recopilar datos relevantes y contextualizados, se llevará a cabo una encuesta en línea a una muestra de 68 desarrolladores de software y líderes de proyectos con experiencia en el desarrollo web, en el uso de metodologías ágiles y conocimiento de ITIL. La encuesta, compuesta por 32 preguntas, incluirá una combinación de preguntas cerradas (escalas de Likert, opciones múltiples) y abiertas. Las preguntas cerradas permitirán cuantificar variables como la frecuencia de uso de diferentes lenguajes de programación y la importancia otorgada a diversos factores. Por otro lado, las preguntas abiertas facilitarán la exploración de las experiencias y opiniones de los participantes en relación con la elección de lenguajes y su impacto en la gestión de proyectos. Los datos obtenidos serán analizados mediante software estadístico Excel, Knime para identificar patrones, tendencias y relaciones entre las variables (Hernández Sampieri & Fernández-Collado, 2014).

Para garantizar que la técnica cumpla con su propósito, será fundamental traducir los objetivos generales y específicos de la investigación en objetivos operativos, los cuales servirán como guía para diseñar las preguntas y orientar la interacción con los participantes (Folgueiras Bertomeu, 2016)

### **3.7.2 Instrumentos**

El instrumento elaborado para esta investigación puede encontrarse en los anexos.

### **3.7.3 Procedimientos**

Para garantizar la obtención de datos válidos y confiables, se seguirá el siguiente procedimiento:

- **Diseño del Instrumento de Recolección:** Se diseñará una encuesta en línea, compuesta por preguntas cerradas y abiertas, que abarque las variables de interés identificadas en el marco teórico. La encuesta será sometida a una prueba piloto para evaluar su claridad y pertinencia.

- **Selección de la Muestra:** Se seleccionará una muestra de 68 profesionales del desarrollo de software, a través de muestro por conveniencia y cuotas, asegurando la diversidad en cuanto a experiencia, roles y empresas.
- **Aplicación de la Encuesta:** La encuesta se enviará a los participantes seleccionados a través de Microsoft Forms junto con una carta de presentación que explique los objetivos de la investigación y garantice la confidencialidad de las respuestas.
- **Recolección de Datos:** Se establecerá un plazo para la respuesta y se realizará un seguimiento para asegurar una alta tasa de respuesta.
- **Limpieza y Codificación de Datos:** Los datos recolectados serán revisados para identificar y corregir errores, inconsistencias o valores atípicos. Posteriormente, se codificarán los datos cualitativos para facilitar su análisis.
- **Análisis de Datos:**
  - **Análisis Cuantitativo:** Se utilizará el software estadístico Excel y Knime para realizar análisis descriptivos (frecuencias, porcentajes) y demás análisis para identificar patrones entre las similitudes.
  - **Análisis Cualitativo:** Los datos cualitativos obtenidos de las preguntas abiertas serán codificados y categorizados utilizando técnicas de análisis de contenido.
- **Interpretación de Resultados:** Los resultados obtenidos serán interpretados a la luz del marco teórico y los objetivos de investigación. Se establecerán relaciones entre los hallazgos y la literatura existente.
- **Elaboración del Informe:** Se elaborará un informe final que presente de manera clara y concisa los resultados de la investigación, incluyendo tablas, gráficos y citas bibliográficas.

### 3.7.4 Plan de análisis

El siguiente plan de análisis detalla los pasos y recursos necesarios que se utilizaran para analizar la información.

**Tabla 13 Plan de análisis**

Fase	Descripción	Herramientas	Objetivo
Recolección de Datos	Diseño y aplicación de encuesta en línea	Microsoft Forms,	Obtener datos cuantitativos y

			cualitativos sobre las percepciones y prácticas de los desarrolladores.
<b>Limpieza y Procesamiento de Datos</b>	Verificación, codificación, transformación de datos	Excel, Knime, Python	Garantizar la calidad de los datos y prepararlos para el análisis.
<b>Análisis de Datos</b>			
<b>* Descriptivo</b>	Cálculo de frecuencias, porcentajes, medidas de tendencia central y dispersión	Knime, Python	Describir las características de la muestra y las variables.
<b>Visualización de Datos</b>	Creación de gráficos, tablas y diagramas	Excel, Power BI	Presentar los resultados de manera clara y concisa.
<b>Interpretación de Resultados</b>	Relación de los resultados con el marco teórico y los objetivos de investigación	Word, PowerPoint	Responder a las preguntas de investigación y generar conclusiones.

Fuente: Elaboración propia.

### 3.8 Fuentes de Información

#### 3.8.1 Fuentes Primarias

Esta investigación se basará principalmente en **datos primarios** recolectados a través de una encuesta en línea dirigida a profesionales del área de software. La decisión de priorizar las fuentes primarias se justifica por la necesidad de obtener información actualizada y contextualizada sobre las prácticas y percepciones de los desarrolladores en relación con la elección de lenguajes de programación. Además, la escasez de estudios previos que aborden específicamente la relación entre la elección de lenguajes y la gestión del ciclo de vida del servicio según ITIL 4 justifica la recolección de datos originales.

Los datos primarios obtenidos permitirán realizar un análisis profundo y detallado de las variables de interés, generando resultados novedosos y relevantes para el campo de la ingeniería de software.

#### 3.8.2 Fuentes Secundarias

La presente no hará uso de datos de fuentes secundarias debido a la naturaleza de la investigación y a la necesidad de recolectar datos de primera mano de expertos.

### 3.9 Juicio de expertos

Esta técnica es utilizada para validar ciertos aspectos de un instrumento de recolección de datos, entre ellas tenemos la pertinencia, la calidad, claridad de preguntas. En este enfoque los

expertos emiten opiniones basadas en su experiencia.

El instrumento utilizado para llevar a cabo esta técnica puede visualizarse en la sección de Anexos dentro del ítem 2.

### 3.10 Resultados de pruebas piloto

Para evaluar la confiabilidad de las pruebas piloto y garantizar la consistencia interna de los ítems que las componen, se empleará la técnica alfa de Cronbach, una técnica estadística ampliamente validada en la metodología de investigación. Este método cuantifica el grado en que los elementos de un instrumento miden un mismo constructo o dimensión, proporcionando como resultado un valor entre 0 y 1, donde valores cercanos a 1 indican alta consistencia interna.

Para realizar el cálculo se utiliza la siguiente formula:

$$a = \frac{K}{K - 1} \left[ 1 - \frac{\sum Vi}{Vt} \right]$$

Donde,

- a = Alfa de Cronbach
- K = Número de ítems
- Vi = Varianza de cada ítem
- Vt = Varianza del total de ítems

Para ello se decidió utilizar un script en Python para el cálculo de este índice, dando como resultado: 0.78, el cual es un valor aceptable alto que fortalece el instrumento de recolección de datos realizado. El script utilizado de Python para este proceso se comparto a continuación:

```
import pandas as pd
from pingouin import cronbach_alpha
ruta_archivo = "/content/cronbach.xlsx"
df = pd.read_excel(ruta_archivo, sheet_name="hoja1")
items = df.filter(like='Item')
alpha, _ = cronbach_alpha(data=items)
print(f"Alfa de Cronbach: {alpha:.3f}")
```

En este script se hace uso de la librería pingouin, la cual es famosa por su cantidad de formular estadísticas que contiene, en este caso se utilizó la función cronbach\_alpha.

### 3.11 Matriz de congruencia

Nombre de la investigación	Problema	Preguntas de investigación	Objetivos de la investigación	Metodología	Instrumentos	Variables	Indicadores
<b>La elección de lenguajes de programación en el desarrollo web</b>	La cantidad de lenguajes que hay hoy en día provoca desorganización internamente en las empresas al tener un abanico de posibilidades tecnológico amplio, obligándolos a desarrollar de forma rápida sin tener la oportunidad de escoger de forma sensata un lenguaje.	<b>General</b> ¿Qué factores, criterios técnicos y contextuales influyen en las preferencias de los profesionales del software al seleccionar lenguajes de programación para el desarrollo web, y cómo se manifiestan estas preferencias según su experiencia y rol?	<b>General</b> Diagnosticar las preferencias de los profesionales del software en el contexto del desarrollo web para comprender los criterios que guían la elección de lenguajes de programación, con el propósito de identificar factores determinantes en dicha selección.	Cualitativa, Cuantitativa, Descriptiva, Exploratoria, Orientada a conclusiones	Encuesta en línea	<b>Independientes</b> Rol Profesional, Conocimiento de ITIL, Aceptación de rúbrica  <b>Dependientes</b> Años de experiencia profesional, Años de experiencia en el desarrollo web, Metodologías en cascada	Rol profesional, Herramientas de gestión de proyectos, Escala de factores técnicos relevantes, Escala de factores empresariales, Marco ITIL, Experiencia profesional, Rúbrica

Nombre de la investigación	Problema	Preguntas de investigación	Objetivos de la investigación	Metodología	Instrumentos	Variables	Indicadores
<b>La elección de lenguajes de programación en el desarrollo web</b>	Lo cual dificulta la gestión de proyectos de desarrollo y puede tener una influencia negativa en la empresa	<b>Especifica</b>  1. ¿Cuáles son los factores determinantes que influyen en la selección de un lenguaje de programación?	<b>Específico</b>  1. Identificar los 5 principales factores que influyen en la selección de un lenguaje de programación entre profesionales del área de desarrollo de software.  2. Diseñar una rúbrica compuesta de los factores identificados que sirva como herramienta de orientación para los responsables de desarrollo en empresas al momento de seleccionar un lenguaje de programación.			Metodologías ágiles, Factores relevantes, Escala de importancia de factores, Considera la sostenibilidad y mantenibilidad, Propósito del proyecto, Paquetes y librerías, Experiencia del equipo de desarrollo, Rubro de la empresa, Factores técnicos del lenguaje	

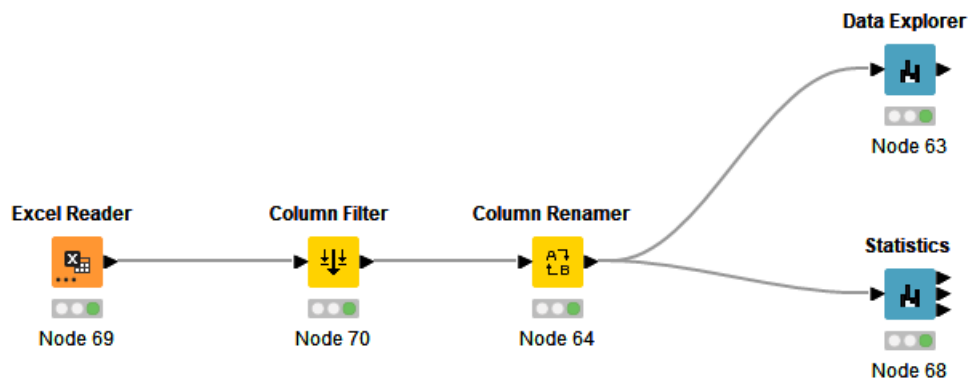
Nombre de la investigación	Problema	Preguntas de investigación	Objetivos de la investigación	Metodología	Instrumentos	Variables	Indicadores
<p><b>La elección de lenguajes de programación en el desarrollo web</b></p>		<p><b>Especifica</b></p> <p>2. ¿Qué criterios pueden utilizarse para construir una rúbrica que facilite la selección de lenguajes de programación?</p>				<p>Fase de ciclo de vida y su relevancia con ITIL, Alineación de objetivos y de ITIL con TI, Experiencia donde la elección del lenguaje fue clave, Participación en proyectos que se han tornado complejos por la elección</p>	
						<p>Nivel de aceptación de rúbrica.</p>	

## CAPITULO IV - RESULTADOS Y ANALISIS

El presente capítulo tiene como objetivo presentar los resultados y el análisis del instrumento de recolección de datos aplicado, el cual permite examinar el marco de la investigación en cuestión. La encuesta, diseñada específicamente para este estudio, fue implementada con el propósito de recopilar información relevante de expertos en el área. Sus respuestas no solo facilitan la interpretación de los datos, sino que también permiten identificar patrones, puntos de coincidencia y similitudes que contribuyen significativamente al desarrollo del estudio. A través de este análisis, se busca obtener una comprensión más profunda y detallada de los aspectos clave abordados en la investigación.

Para el análisis de datos, se empleará inicialmente un Análisis Exploratorio de Datos (EDA), el cual incluye una serie de técnicas propias de la estadística descriptiva, como el cálculo de medias, medianas, desviaciones estándar y frecuencias. Estas herramientas permitirán resumir y cuantificar las características principales de los datos, ofreciendo una visión clara y estructurada de la información recopilada. Además, el EDA servirá como base para detectar tendencias, anomalías o relaciones entre variables, lo que enriquecerá la interpretación de los resultados y fortalecerá las conclusiones del estudio.

### 4.1 Análisis Exploratorio de los datos



#### Ilustración 6 Estructura conocimientos generales

Fuente: Elaboración Propia

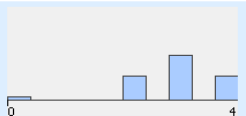
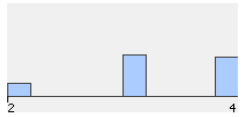
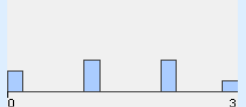
##### 4.1.1 Análisis de conocimientos generales

Para el EDA de la primera sección se utilizaron los nodos de la ilustración 6, cabe recalcar

que no se encontraron valores nulos en esta sección por lo que no fue necesario realizar un tratamiento de datos para corregirlos.

Los resultados sobre estos datos son los siguientes:

**Tabla 14 Análisis de conocimientos generales**

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
Metodologías Cascada	0.0	2.9167	3	4	0.8681	-0.8988	1.7882	0	0	0	
Metodologías Agiles	2	3.2778	3	4	0.6965	-0.4393	-0.8511	0	0	0	
ConocimientoITIL	0.0	1.3333	1	3	0.9494	0.0903	-0.9178	0	0	0	

Fuente: Elaboración propia

Estos datos reflejan que la media de las metodologías ágiles es superior a la de metodologías en cascadas, lo cual sugiere que en cuanto a gestión de proyectos en la industria del software suele utilizarse mayormente por sobre las metodologías tradicionales o en cascada. Este hallazgo es consistente con la tendencia actual en la industria, donde las metodologías ágiles son valoradas por su flexibilidad y adaptabilidad.

Sobre ITIL, se muestra una media baja, lo cual indica que muchos participantes cuentan con un conocimiento nulo o escaso de esta herramienta; esto puede estar precedido por la cantidad de desarrolladores que participaron en la encuesta, lo que demuestra que ITIL es utilizado en la industria por roles más altos al de un desarrollador. En cuanto al sesgo con valor de 0.09 indica una tendencia a valores bajos recalcando el bajo conocimiento sobre esta herramienta.

#### 4.1.2 Análisis sobre ITIL

En el análisis realizado, se siguió un esquema similar al presentado en la ilustración 6, el cual, por razones de espacio o relevancia, no se incluye en esta sección. Dado que el conjunto de datos utilizado es de naturaleza cualitativa, se consideró esencial aplicar un tratamiento especial para transformar estos datos en cuantitativos, lo que permite un análisis más robusto y compatible con técnicas estadísticas y modelos matemáticos.

Además, durante la revisión del conjunto de datos, se identificó la presencia de valores

nulos o faltantes. Para manejar esta situación, se decidió imputar estos valores con el número 0 en el caso de variables numéricas, y con el carácter "0" en el caso de variables de texto.

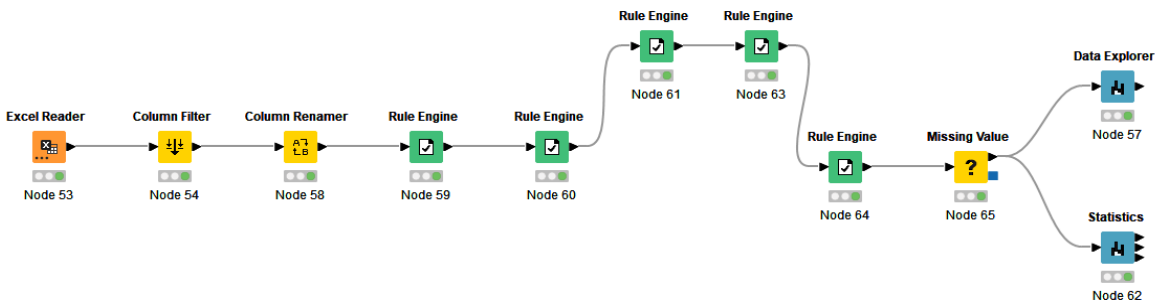
Los datos nulos se dejan evidenciados en la siguiente tabla.

**Tabla 15 Valores nulos**

...	S Rol pro...	D Proyectos con ITIL	S ▲ Cido Vida ITIL	I Alineaci...	I SLA ITIL	I Influen...	I Entreg...
..	Desarrollador	0	?	0	0	0	1
..	Desarrollador	0	?	0	0	0	1
..	Arquitecto d...	1	Diseño del servicio	3	2	3	3
..	Líder de pro...	1	Diseño del servicio	4	4	4	4
..	Arquitecto d...	0	Diseño del servicio	3	2	4	3
..	Desarrollador	0	Diseño del servicio	0	0	1	2
..	Líder de pro...	0	Diseño del servicio	2	1	3	2

Fuente: Elaboración propia

Luego de aplicar las reglas para el tratamiento de datos y para evitar los datos nulos el diagrama final puede verse en la ilustración 7.



**Ilustración 7 Estructura tratamiento de datos ITIL**

Fuente: Elaboración propia

El resultado de los datos muestra valores sorprendentes en cuanto al uso de ITIL en este rubro del desarrollo de software, en cuanto a proyectos de ITIL otorga una media muy baja de 0.27. Esto indica que, a pesar de que ITIL es un marco de trabajo diseñado específicamente para la gestión de servicios de TI, su adopción en este rubro es mínima. La baja participación de ITIL en proyectos de desarrollo de software sugiere una brecha entre las prácticas de gestión de servicios de TI y las necesidades específicas del desarrollo de software. Esto podría representar una oportunidad para mejorar la integración de ITIL en este ámbito.

En cuanto a la alineación de ITIL con los objetivos de la empresa los resultados presentan que la mayoría de las respuestas están en niveles intermedios, pero con un sesgo bajo de 0.20, lo cual indica que hay una tendencia a niveles bajos. Esto indica que ITIL no está siendo aprovechado.

al máximo para apoyar los objetivos de la empresa. Esto resalta la necesidad de una mejor comunicación, adaptación y capacitación en torno al uso de esta herramienta. La desviación estándar es de 1.3082, la cual es moderadamente alta e indica variabilidad en cómo los objetivos están alineados con los proyectos. La curtosis es de -1.0738 indicando una distribución más plana que una distribución normal.

La influencia de ITIL en las empresas tiene una media y mediana de 2 lo que sugiere una distribución asimétrica. En cambio, la desviación estándar tiene un valor de 1.5198.

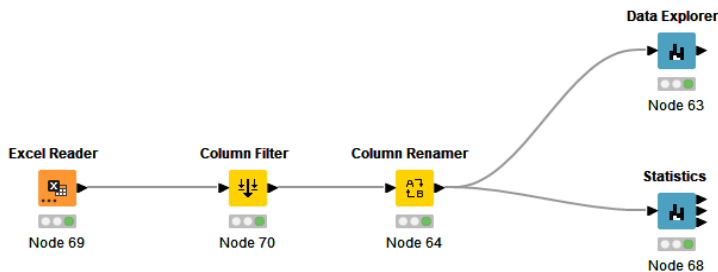
**Tabla 16 Análisis ITIL**

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
Proyectos con ITIL	0.0	0.2778	0.0	1	0.4187	1.0042	-0.807	0	0	0	
Alineacion Objetivos ITIL	0.0	1.5833	2	4	1.3082	0.2012	-1.0738	0	0	0	
SLA ITIL	0.0	1.4167	2	4	1.1476	0.1526	-0.9755	0	0	0	
Influencia ITIL Empresa	0.0	2	2	4	1.5198	-0.0495	-1.4879	0	0	0	
Entregas Rapidas	0.0	2.3611	2	4	0.7927	-0.399	1.1273	0	0	0	

Fuente: Elaboración propia

### 4.1.3 Análisis sobre factores técnicos

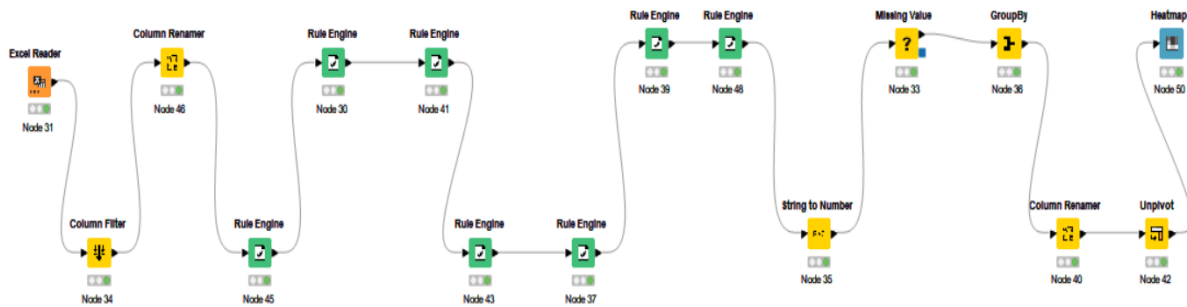
El siguiente análisis se basó exclusivamente en factores técnicos del desarrollo de software, el objetivo principal es conocer las facetas que mayor relevancia generan al momento de elegir y los elementos que son considerados al escoger un lenguaje de programación.



**Ilustración 8 Estructura factores técnicos**

Fuente: Elaboración propia

Sin embargo, muchos de los datos eran valores cualitativos, por ende, se realizó un tratamiento especial para convertir los valores de campos como el conocimiento de ITIL y rol profesional a datos cuantitativos. Posteriormente se identificaron datos nulos en varias columnas del set de datos, para lo cual se incluyeron elementos que permitan convertir estos datos, la estructura final puede observarse en la ilustración 9.



**Ilustración 9 Estructura factores técnicos**

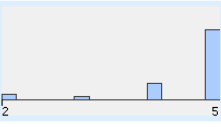
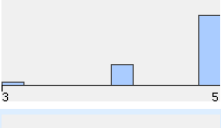
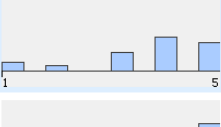
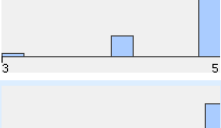
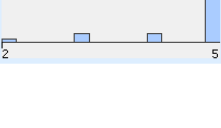
Fuente: Elaboración propia

Este análisis muestra que la elección de un lenguaje de programación esta influenciada por varios factores, sin embargo, hay algunos que fueron muy votados en comparación a otros, los resultados pueden generales pueden verse en la siguiente tabla.

**Tabla 17 Análisis factores técnicos**

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
Sostenibilidad	0.0	0.8056	1	1	0.3202	-1.4238	0.8622	0	0	0	
Proposito Proyecto	0.0	3.5278	4	5	1.0743	-1.7564	4.4724	0	0	0	
Experiencia Equipo	0.0	0.875	1	1	0.2751	-2.1517	3.7062	0	0	0	

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
Abundancia Librerías	0.0	0.9444	1	1	0.1978	-3.7903	14.3714	0	0	0	
rubroEmpresa	0.0	0.6111	0.5	1	0.413	-0.4392	-1.3975	0	0	0	
Paradigmas de programación	2	4.3333	4	5	0.7506	-1.0506	0.9896	0	0	0	
Patrón de diseño de programación	2	4.1667	4	5	0.8392	-0.621	-0.5196	0	0	0	
Multitasking	1	4.1944	4	5	0.9137	-1.5402	3.0153	0	0	0	
Nullsafety	1	2.8889	3	5	1.338	-0.0093	-1.0754	0	0	0	
proyectosDificultanXLenguajes	0.0	0.5	0.5	1	0.5035	-2.52E-17	-2.058	0	0	0	
Infraestructura	0.0	0.7778	1	1	0.4187	-1.3649	-0.1417	0	0	0	
Soporte y comunidad	3	4.6111	5	5	0.5453	-0.995	-0.0324	0	0	0	
Popularidad	2	4.7222	5	5	0.6965	-2.6475	6.4426	0	0	0	
Rendimiento del lenguaje	1	4.6944	5	5	0.7807	-3.4178	13.0355	0	0	0	
Propósito del proyecto	3	4.0278	4	5	0.731	-0.0431	-1.0944	0	0	0	
Soporte multiplataforma	1	4.25	5	5	1.0448	-1.4386	1.4729	0	0	0	

Column	Min	Mean	Median	Max	Std. Dev.	Skewness	Kurtosis	No. Missing	No. +∞	No. -∞	Histogram
Seguridad del lenguaje	2	4.6111	5	5	0.7971	-2.2637	4.5503	0	0	0	
Compatibilidad con otros sistemas	3	4.7222	5	5	0.5097	-1.6506	1.9266	0	0	0	
Necesidades del cliente	1	3.75	4	5	1.1956	-0.9217	0.1605	0	0	0	
Abundancia de librerías y paquetes	3	4.7222	5	5	0.5097	-1.6506	1.9266	0	0	0	
Facilidad de aprendizaje	2	4.6667	5	5	0.7506	-2.2382	4.0801	0	0	0	

Fuente: Elaboración Propia

Sobre la seguridad del lenguaje, la media es alta con un valor de 4.72 indicando la alta consideración que tienen los profesionales por este aspecto. El sesgo es negativo (-2.2637), indicando que la mayoría de las respuestas están en la parte superior de la escala.

La compatibilidad con otros sistemas se muestra como un factor importante, la media es de 4.72, el sesgo es moderado de -1.6506 y la curtosis es de 1.9266 confirmando así que la mayoría de las respuestas se encuentran con valoración altas para este factor.

La abundancia de librerías es presentada como un factor crucial, prueba de ello es la media alta de 4.72 y un sesgo negativo relativamente menor de -1.6506 lo cual indica la prevalencia de valores altos por sobre valores bajos.

Caso contrario con sobre las necesidades del cliente, cuyo factor parece ser determinante según contextos y el alcance del proyecto, ya que cuenta con una de las medias más bajas de todos los factores presentados en la encuesta (3.75), con una dispersión dispersa de 1.1956 confirmando que puede ser tomado en cuenta en proyectos específicos.

En forma de resumen, los resultados con mayor media pueden ser vistos en la tabla a continuación.

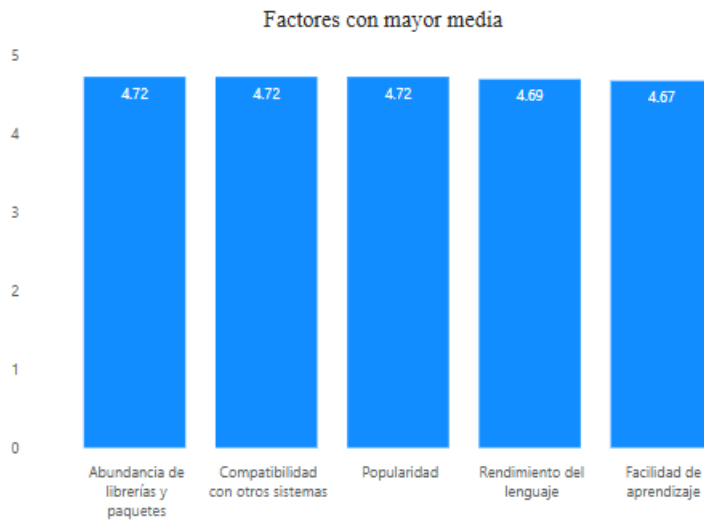
**Tabla 18 Factores elegidos mayor a menor**

Factor	Media (0- 5)
Compatibilidad con otros sistemas	4.72
Abundancia de librerías y paquetes	4.72
Popularidad	4.72
Rendimiento del lenguaje	4.69
Facilidad de aprendizaje	4.67
Soporte y comunidad	4.61
Seguridad del lenguaje	4.61
Paradigmas de programación	4.33
Soporte multiplataforma	4.25
Multitasking	4.19
Patrón de diseño de programación	4.17
Propósito del proyecto	4.03
Rubro de la empresa	3.83
Necesidades del cliente	3.75
Propósito Proyecto	3.53
Nullsafety	2.89
Abundancia Librerías	0.94
Experiencia Equipo	0.88
Sostenibilidad	0.81
Infraestructura	0.78
Rubro de la Empresa	0.61

Fuente: Elaboración propia

Estos datos muestran una clara preferencia por factores netamente técnicos y propios del desarrollo de software, tal y como es la compatibilidad con otros sistemas y la abundancia de librerías. Sin embargo, en el top 3 se encuentra la popularidad del lenguaje como un factor a considerar, esto puede estar incitado a qué en la medida que un lenguaje es más popular significa que recibirá soporte prolongado y existirá una comunidad activa.

El rendimiento de lenguaje fue un factor valorado, lo cual indica que se busca lenguajes que ofrezcan alta eficiencia, esto es clave en sistemas de la actualidad en los cuales el tiempo real, la velocidad de respuesta y el procesamiento de grandes volúmenes de información es clave para garantizar eficiencia y velocidad en los sistemas informáticos.



### **Ilustración 10 Factores con mayor media**

Fuente: Elaboración propia

Sobre los datos con menor medida, tenemos que la experiencia y conocimiento previo del equipo en ciertos lenguajes es factor para considerar, sin embargo y a pesar de que la experiencia puede influir en la curva de aprendizaje de un lenguaje, su baja puntuación sugiere que los desarrolladores priorizan otros aspectos, como la eficiencia y compatibilidad del lenguaje, por encima de la familiaridad del equipo con una tecnología específica.

En cuanto a la infraestructura, el cual es un factor con una media baja sugiere que no es una barrera significativa en la adopción de un lenguaje. Esto puede deberse a la versatilidad de herramientas modernas y flexibles como Docker, el cual disminuye considerablemente este aspecto.

El rubro de la empresa fue el elemento menos seleccionado, lo cual demuestra que su influencia es mínima en este aspecto y no es considerado como un limitante. Esto sugiere que no existen reglamentos gubernamentales que regulen el uso de un lenguaje en un sector en específico, por ende, una empresa es libre del lenguaje a utilizar según su conveniencia.

## **4.2 Informe del proceso de recolección de datos**

### **4.2.1 Descripción del proceso**

El proceso de recolección de datos se llevó a cabo en tres etapas principales: diseño, aplicación y análisis. En la primera etapa, se diseñó la encuesta en línea utilizando la plataforma Microsoft Forms, la cual permitió estructurar las preguntas de manera clara y accesible.

En la segunda etapa se distribuyó la encuesta a las personas seleccionadas para formar parte del estudio y finalmente se procedió a la depuración y posterior análisis de los datos.

#### 4.2.2 Participantes

La muestra estuvo conformada por un grupo de expertos que fueron seleccionados bajo los criterios de inclusión y exclusión, los cuales están definidos en el capítulo anterior.

#### 4.2.3 Instrumento

El instrumento seleccionado fue una encuesta en línea compuesta por 32 preguntas y divididas en secciones relevantes y de interés. La estructura de la encuesta puede verse en los anexos.

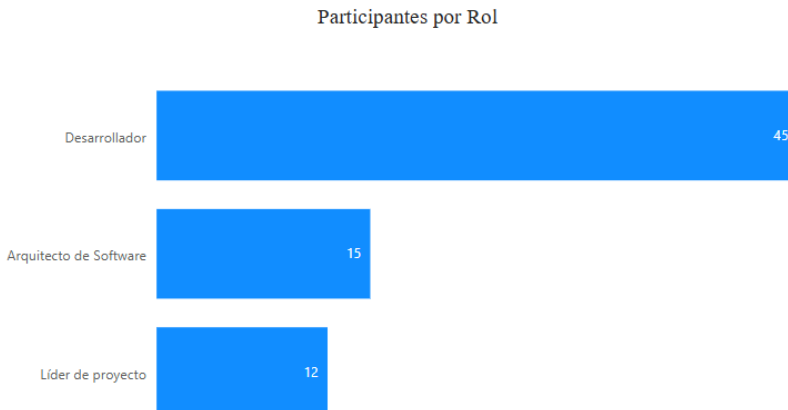
#### 4.2.4 Dificultades encontradas

Al ser un estudio dirigido a un grupo específico de la rama que suele asociarse a Ingenieros en Sistemas, Computación, Informáticos y demás carreras relacionadas; al principio de la recolección se ha encontrado una baja tasa de respuesta inicial que poco a poco ha subido considerablemente.

#### 4.2.5 Consideraciones éticas

Se garantizó la confidencialidad de los participantes asegurando que los datos recopilados fueran única y exclusivamente con el fin académico de concluir el presente trabajo investigativo. Esto les fue informado desde un principio además de estar presente en el instrumento en el encabezado haciendo referencia a la privacidad y no publicación de la información recopilada.

#### 4.2.6 Generalidades de los datos obtenidos



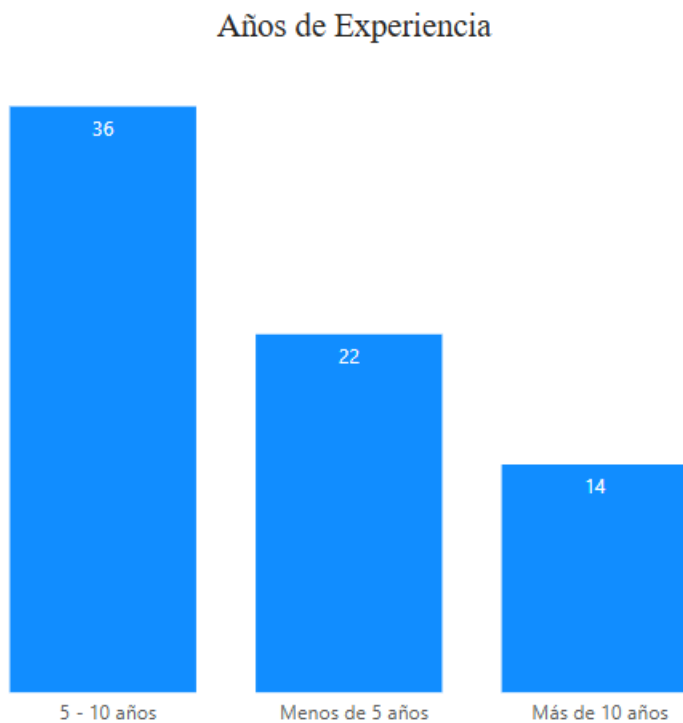
#### Ilustración 11 Rol Profesional

Fuente: Elaboración propia

El primer aspecto importante para analizar es que la mayoría de las personas que llenaron

la encuesta tienen el rol de desarrolladores de software en sus empresas. El 62.50% corresponde a desarrolladores, el 20.83% corresponde a arquitectos de software y el 16.67% compete a líderes de proyecto.

Evaluando los años de experiencia profesional de los participantes se observa un promedio significativo de personas que tienen entre 5 y 10 años de experiencia, su valor es de 36, mientras tanto para gente con experiencia de menos de 5 años el valor es de 22 y para la gente con experiencia de más de 10 años el porcentaje se reduce a una cantidad de 14.



**Ilustración 12 Años de Experiencia Profesional**

Fuente: Elaboración propia

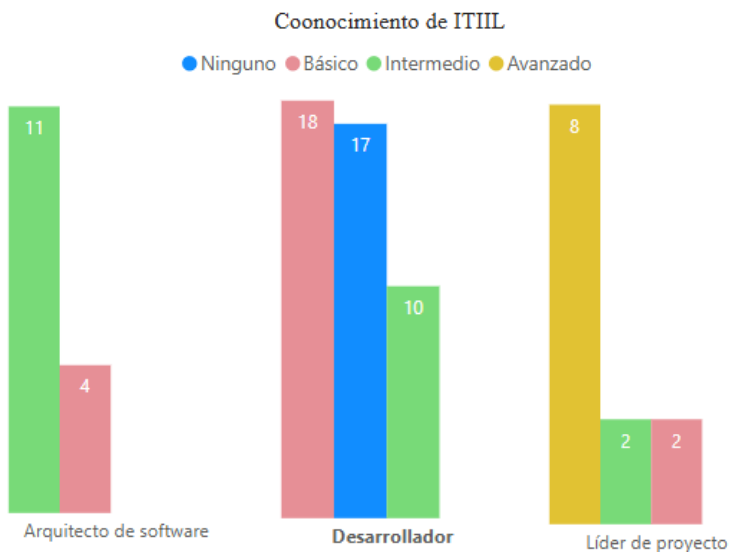
Sobre ITIL se observa un comportamiento interesante en los resultados de la encuesta. El grupo predominante entre los encuestados es el de desarrolladores, lo cual podría sugerir que, en el contexto de la organización o sector analizado, este perfil tiene una presencia significativa en proyectos relacionados con la gestión de servicios de TI. Sin embargo, un hallazgo llamativo es que ningún desarrollador afirma tener un conocimiento avanzado de ITIL, según se refleja en el gráfico presentado. Esto podría deberse a que, aunque los desarrolladores interactúan con procesos de TI, su enfoque principal está más orientado hacia la creación de software y la resolución de

problemas técnicos, lo que posiblemente limita su profundización en marcos de gestión como ITIL.

Por otro lado, el gráfico también revela que los líderes de proyecto son quienes consideran tener un conocimiento avanzado de ITIL. Esta percepción podría explicarse por su constante contacto con metodologías de gestión y su rol estratégico en la organización.

Los líderes de proyecto suelen ser responsables de coordinar equipos, alinear los objetivos del departamento con las metas organizacionales y garantizar que los servicios de TI se entreguen de manera eficiente y alineada con las mejores prácticas. Por lo tanto, es probable que estén más familiarizados con frameworks como ITIL, ya que les proporciona herramientas y procesos estandarizados para mejorar la calidad de los servicios, gestionar incidentes y optimizar la entrega de valor al negocio.

Estos datos sugieren que, aunque ITIL es reconocido como un marco de referencia clave en la gestión de servicios de TI, su nivel de adopción y profundidad de conocimiento varía significativamente según el rol dentro de la organización. Esto abre una oportunidad para fortalecer la capacitación en ITIL, especialmente entre los desarrolladores, con el fin de cerrar la brecha de conocimiento y garantizar una implementación más efectiva y cohesionada de sus principios en todos los niveles de la organización.

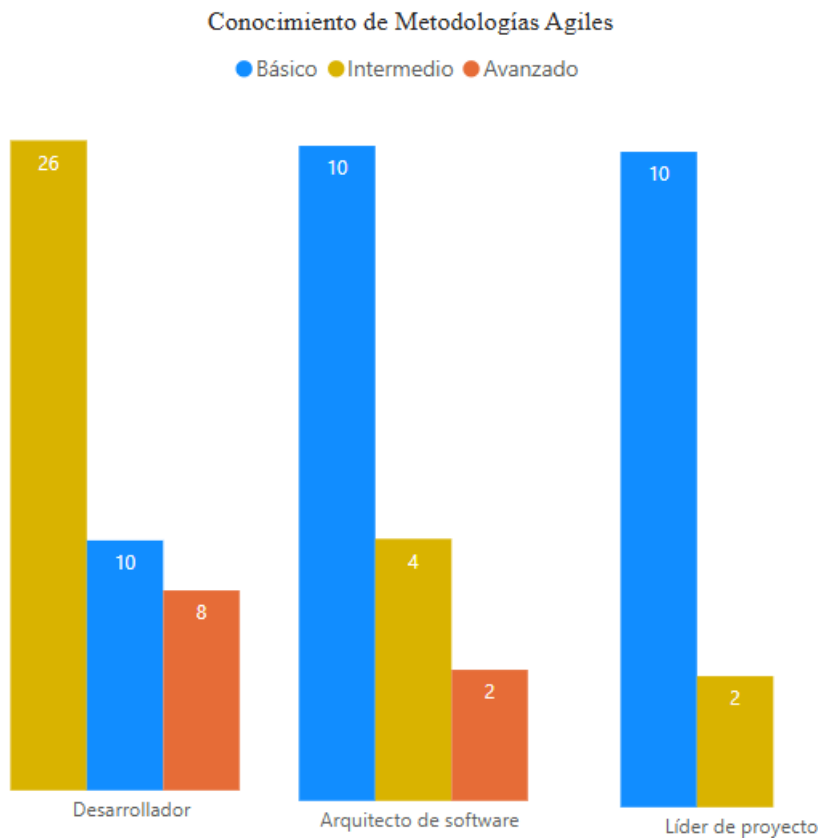


### Ilustración 13 Conocimiento de ITIL

Fuente: Elaboración propia

De igual forma se observa un comportamiento en cuanto al conocimiento de metodologías

agiles, lo cual demuestra que a medida un profesional tiene años de experiencia va expandiendo sus habilidades y busca conocer sobre metodologías de gestión aparte de las requeridas habilidades técnicas que son necesarias para trabajar en el día a día. En el grafico demuestra lo vigente que están las metodologías agiles hoy en día y que prácticamente son utilizadas como elemento clave en el día a día dentro del área puesto que todos tienen un conocimiento sobre esta herramienta.



**Ilustración 14 Conocimiento sobre metodologías agiles para la gestión de proyectos**

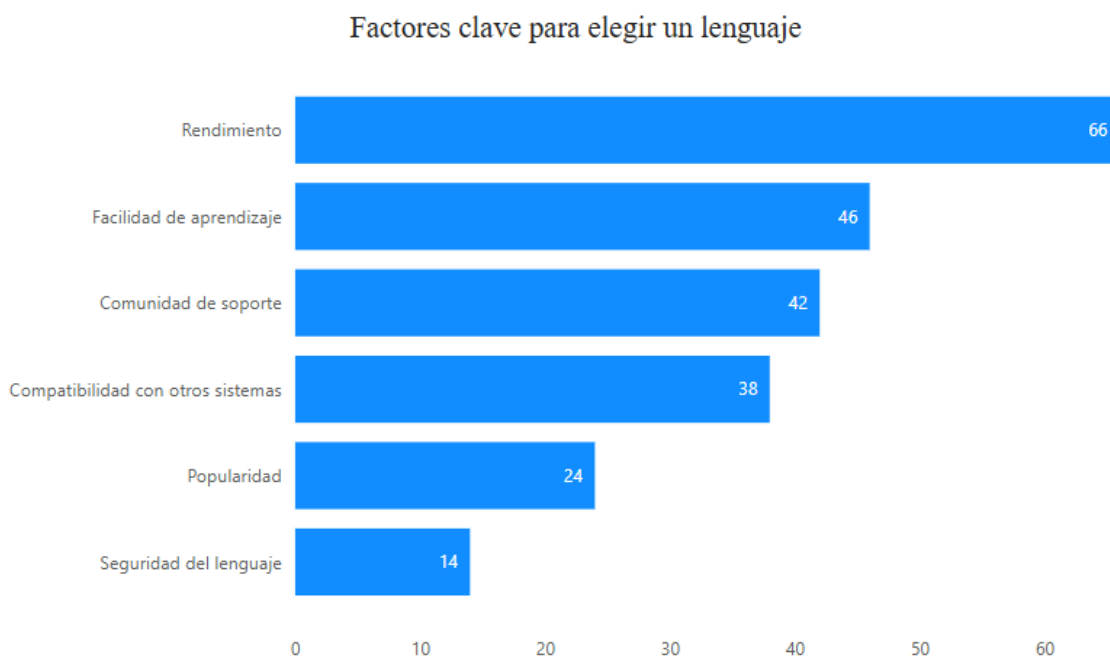
Fuente: Elaboración propia

**4.2.7 Factores técnicos**

Dentro de los elementos considerados clave para el estudio tenemos la pregunta en el instrumento que intenta conocer los elementos que los encuestados perciben como clave para elegir lenguajes de programación. El siguiente grafico nos muestra que la mayoría de las personas considera el rendimiento como el factor que debe prevalecer a la hora de escoger, sin embargo, lo llamativo resulta lo distante que esto se encuentra de las demás opciones.

En resumen, los datos obtenidos y analizados son los siguientes, 66 personas consideran al

rendimiento como elemento principal, seguido de 46 personas que consideran la facilidad de aprendizaje, en tercer lugar, tenemos la comunidad de soporte con 42 personas y la compatibilidad del sistema con un total de 38 elecciones. Rezagados quedan las 2 opciones, como la popularidad y la seguridad del lenguaje con un total de 24 y 14 votos respectivamente. Los datos pueden observarse en la ilustración número 15.



**Ilustración 15 Factores considerados claves para elegir lenguajes de programación**

Fuente: Elaboración propia.

Los resultados obtenidos a la pregunta sobre el nivel de importancia que se le debe dar a la experiencia del equipo de trabajo en cuanto a lenguajes muestran que la mayoría de los votantes consideran que es un factor determinante con 34 votos, en cambio únicamente 2 personas lo consideran para nada determinante y 2 personas también lo consideran poco determinante.

Estos resultados podrían indicar que al contar con equipos de desarrollo es crucial tomar en cuenta los conocimientos y habilidades que ya tienen los colaboradores, ya que esto significa una reducción del tiempo de desarrollo porque se evita la etapa de aprendizaje de un lenguaje y el tiempo de adaptación al mismo. Sin embargo, los resultados también muestran una considerable cantidad de votos sobre que esto no es determinante y desde el punto de vista gerencial, un colaborador debe adaptarse rápidamente a las necesidades y criterios que una empresa disponga.

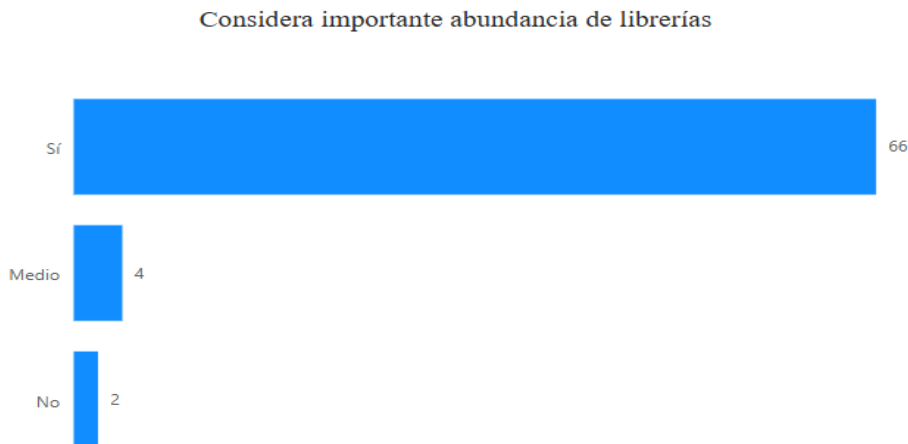
Los resultados pueden observarse en la ilustración 16.



**Ilustración 16 Considera la experiencia un factor para tomar en cuenta**

Fuente: Elaboración propia

Otro factor clave es la abundancia de librerías, donde prácticamente todos los encuestados comentan que es un factor importante a la hora de tomar una decisión. Desde la experiencia profesional esto puede observarse como un punto a favor de no inventar un proceso desde 0 de algo que ya se encuentra realizado salvo necesidad extrema o proyectos demasiados personalizados que requieran un desarrollo muy customizado que dificulte la integración de librerías.



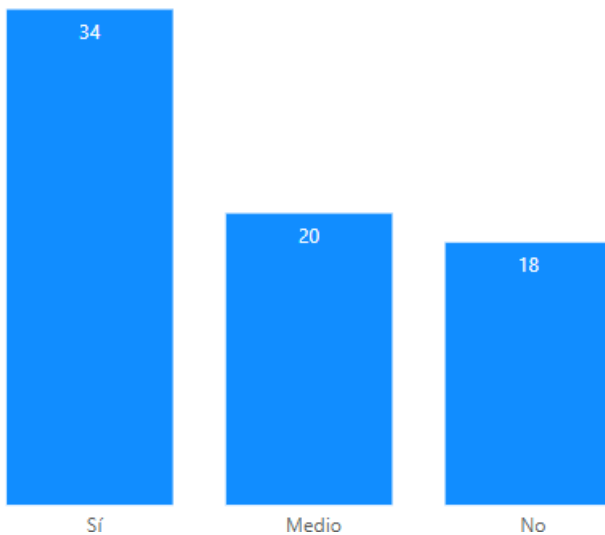
**Ilustración 17 Considera importante la abundancia de librerías**

Fuente: Elaboración propia

Sobre el rubro de la empresa y su influencia a la hora de tomar una decisión sobre lenguajes

de programación se obtuvo como resultado que un total de 34 personas consideran que sí hay influencia directa, 20 personas consideran que puede o no haber influencia y 18 personas consideran que no influye. Estos datos pueden tener como explicación que aquellas personas que trabajan en el rubro financiero o en un rubro manufacturero se ven inducidos y prácticamente obligados a trabajar con un lenguaje específicamente.

Considera influencia del rubro de la empresa

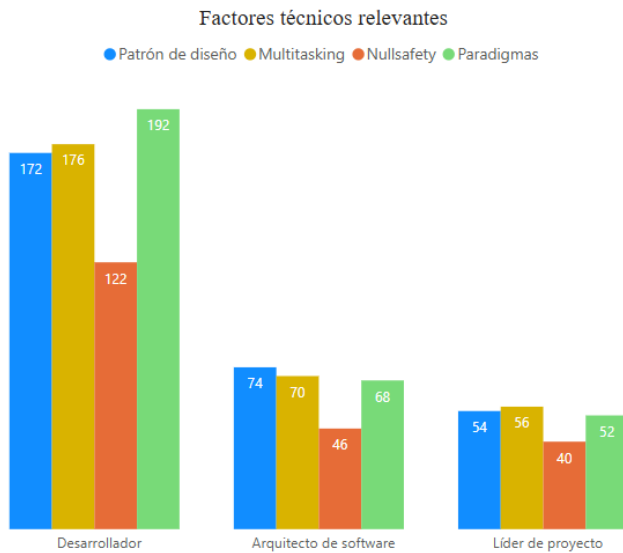


**Ilustración 18 Considera importante el rubro de la empresa**

Fuente: Elaboración propia

Dentro de los factores relevantes, se consultó sobre 4 factores considerados de importancia dentro del desarrollo de software, los participantes eligieron como el de mayor importancia el de paradigmas de programación con el de mayor relevancia, esto se da incluso comparando los resultados de los diferentes roles. En el rol con mayor jerarquía (líder de proyecto) se muestra una distribución entre las opciones con resultados muy parejos, lo cual demuestra la amplia diversidad que existe dentro del rubro y que las opiniones pueden variar según el rol que se tiene en el momento y las labores realizadas en el día a día.

La coincidencia entre los 3 roles radica en que la opción de nullsafety es la que cuenta con menor cantidad de puntos asignados, quedando por debajo de las opciones de paradigma, patrón de diseño y multitasking. El ranking puede observarse en la ilustración 19.

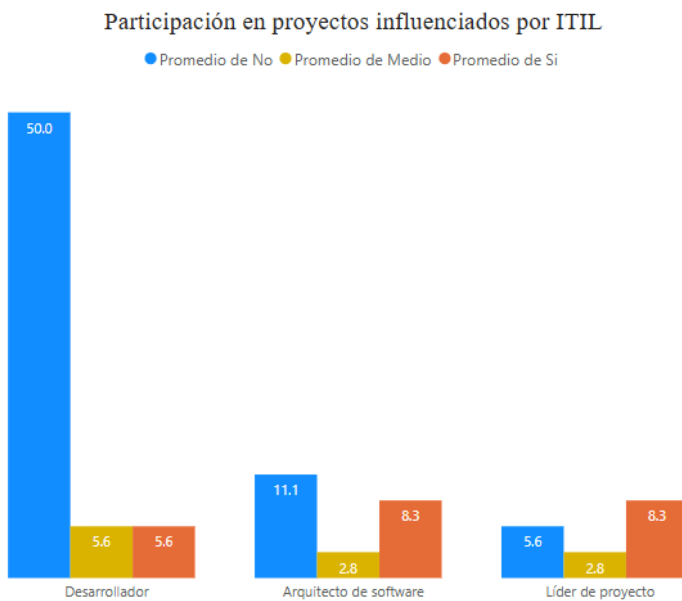


**Ilustración 19 Factores técnicos importantes para elegir un lenguaje**

Fuente: Elaboración propia

#### 4.2.8 Datos obtenidos sobre ITIL

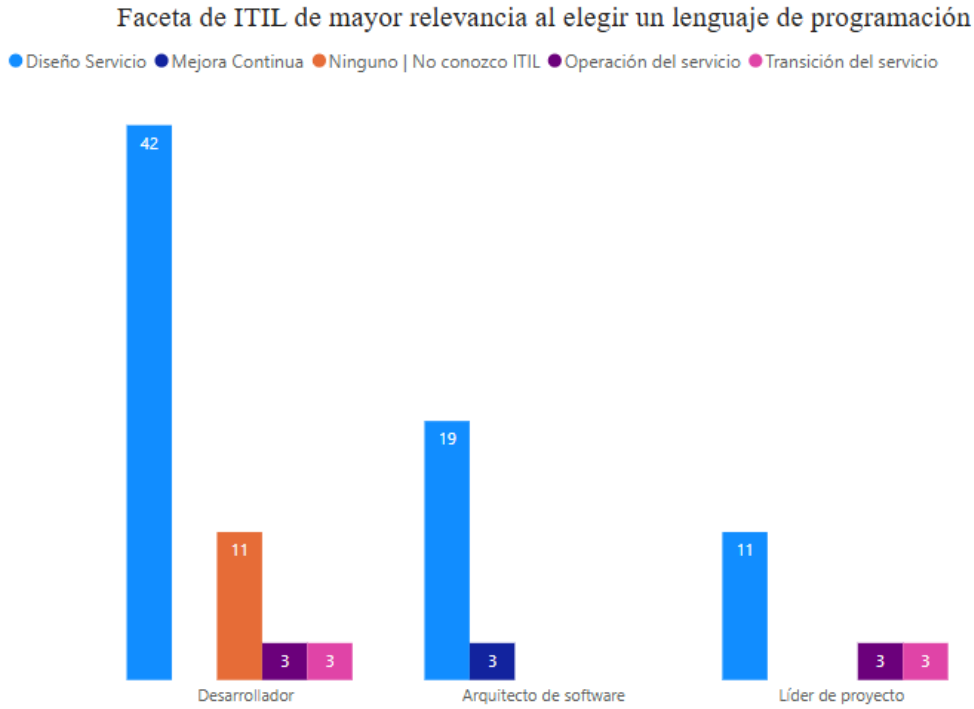
Según los datos recabados, únicamente el 22.2% de los encuestados ha participado en proyectos que estén influenciados por las métricas y prácticas que propone ITIL, hablando del rol de desarrolladores, el 50% no ha participado en este tipo de proyectos y apenas el 5.5% considera que ha participado en proyectos de desarrollo que hayan tenido una influencia de ITIL.



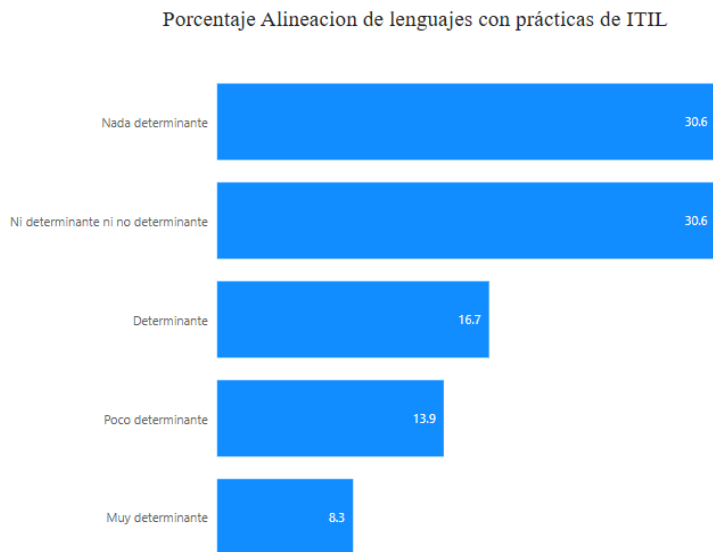
**Ilustración 20 Participación en proyectos influenciados ITIL**

Fuente: Elaboración propia

Ahora, sobre la faceta de ITIL en la cual mayor importancia consideran que tiene un lenguaje de programación, se obtuvo que la mayoría considera que la etapa de diseño del servicio tiene mayor influencia, los datos desglosados por roles pueden verse en la ilustración 21.

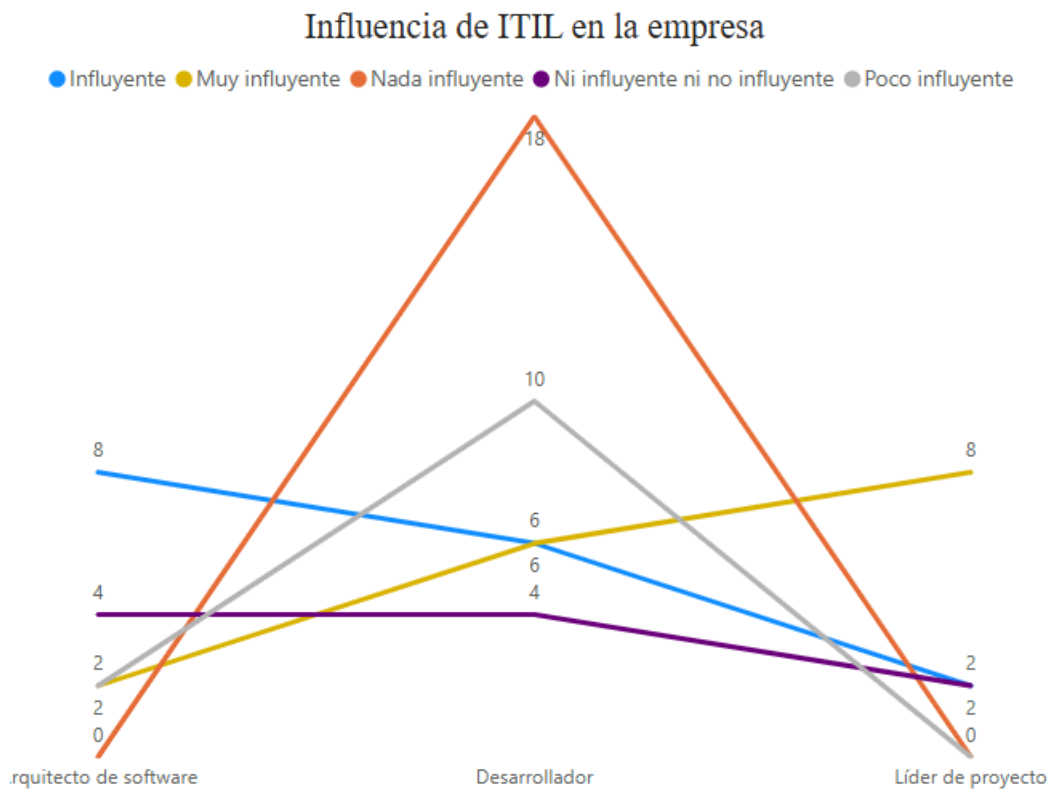


**Ilustración 21 Faceta de ITIL de mayor relevancia al elegir un lenguaje**  
Fuente: Elaboración propia



**Ilustración 22 Alineación lenguajes con prácticas de ITIL**  
Fuente: Elaboración propia

En la evaluación sobre la pregunta del nivel de importancia de la alineación de los lenguajes de programación con los estándares y políticas de ITIL en la empresa para la cual laboran, se identifica que el 30.6% considera que no es nada determinante y únicamente el 8.3% considera que la elección de lenguajes de programación está influenciada por ITIL, los resultados pueden verse en la ilustración 22.



**Ilustración 23 Influencia de ITIL en la empresa**

Fuente: Elaboración propia

El gráfico de la ilustración 23 muestra la percepción de la influencia de ITIL en la empresa según tres roles: arquitecto de software, desarrollador y líder de proyecto. Se observa que los desarrolladores, en su mayoría, consideran que ITIL no tiene ninguna influencia en su trabajo, con un número significativamente alto de respuestas en esa categoría. En contraste, los líderes de proyecto son quienes más perciben a ITIL como una metodología influyente o muy influyente, lo que sugiere que su aplicación es más relevante en la gestión y organización de proyectos.

Por otro lado, los arquitectos de software presentan una distribución más equilibrada, con opiniones variadas sobre el impacto de ITIL. Esto indica que, si bien pueden encontrar utilidad en

sus principios, no lo consideran un factor determinante en su labor diaria. En general, el gráfico refleja que ITIL tiene mayor aceptación en roles de gestión, mientras que en perfiles más técnicos su impacto es percibido como menor o incluso irrelevante.

Sobre la pregunta que intenta identificar el porcentaje de participación en proyectos que se hayan complicado por la elección de un lenguaje se obtuvo que el 50% de los participantes votaron que sí y el restante 50% votaron que no. El gráfico puede observarse en la siguiente ilustración.

Porcentaje de participación en proyectos que se complican por la elección de un lenguaje



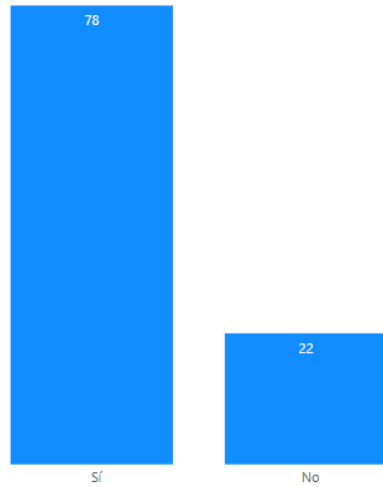
#### **Ilustración 24 Porcentaje de participación en proyectos que se complican por la elección de un lenguaje**

Fuente: Elaboración propia

La pregunta orientada a los costos, diseñada para identificar si los participantes consideran el presupuesto al iniciar un proyecto de software, reveló que el 78% de los encuestados toma en cuenta los costos asociados. Estos costos incluyen aspectos clave como la infraestructura necesaria, las licencias de software y los entornos de desarrollo. Este hallazgo resalta la importancia que los profesionales del desarrollo de software otorgan a la planificación financiera desde las etapas iniciales de un proyecto.

La infraestructura, que puede abarcar desde servidores y almacenamiento en la nube hasta hardware específico, es uno de los factores más mencionados. Los participantes reconocen que una infraestructura adecuada es fundamental para garantizar el rendimiento y la escalabilidad del proyecto.

Considera los costos asociados (infraestructura, licencias) al elegir lenguajes

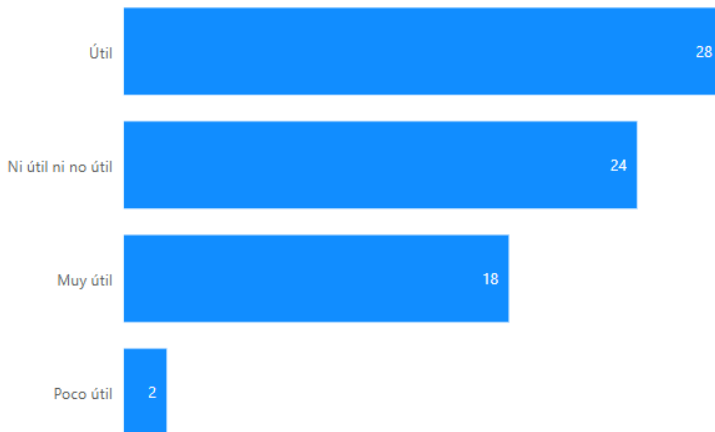


**Ilustración 25 Porcentaje de consideración de costos asociados (infraestructura, licencias)**

Fuente: Elaboración propia

Sobre la pregunta relacionada al nivel de aceptación que tendría una rúbrica que permita determinar una elección correcta sobre un lenguaje de programación, los datos obtenidos reflejan mayormente una aceptación con total de 28 personas que lo consideran útil, la cantidad de personas que votaron por la opción de ni útil ni no útil sobre la rúbrica son un total de 24 personas, sobre la opción de muy útil, esta fue seleccionada por un total de 18 personas y únicamente 2 personas consideran que la opción de una rúbrica es poco útil.

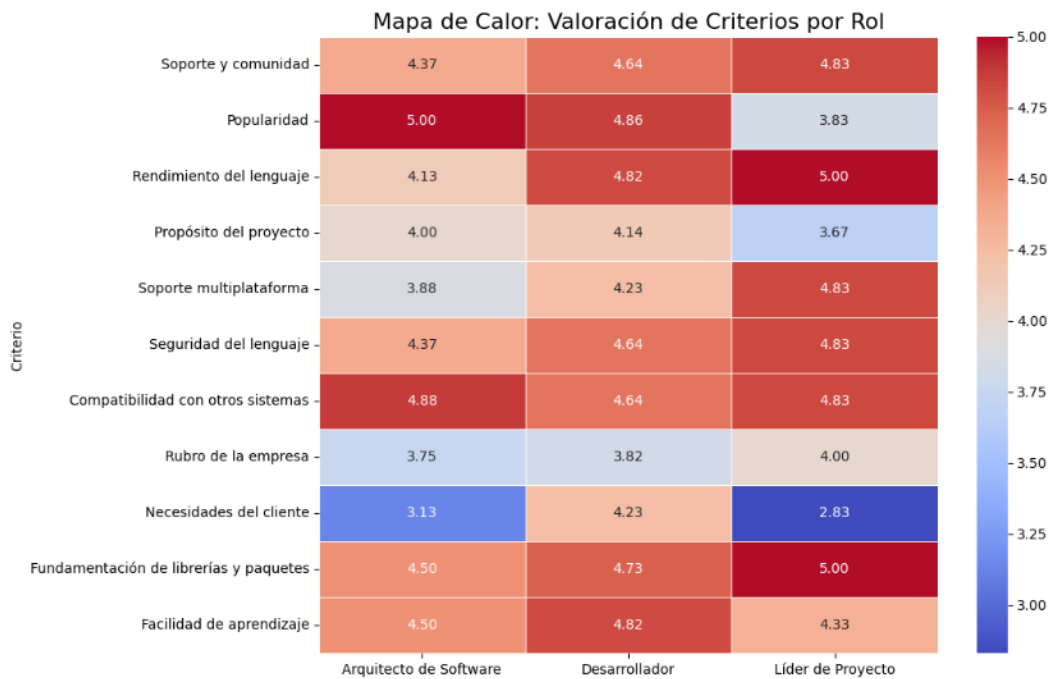
Consideraría útil una rúbrica para seleccionar lenguajes de programación



**Ilustración 26 Aceptación de rúbrica**

Fuente: Elaboración propia

Estas respuestas no solo estuvieron influenciadas por la experiencia personal y profesional de cada participante, sino que también evidenciaron cómo los diferentes roles jerárquicos dentro de la industria condicionan la percepción de lo que es esencial en una rúbrica. Por ejemplo, los desarrolladores tendieron a enfocarse en aspectos más técnicos y específicos, como la facilidad de aprendizaje. Por otro lado, los roles más senior, como arquitectos de software o líderes de equipo, destacaron la importancia de incorporar elementos estratégicos, como la escalabilidad del software, la seguridad del software y el soporte y la comunidad con la que cuenta el lenguaje. Los datos pueden ser vistos en el mapa de calor que corresponde a la ilustración 27.



**Ilustración 27 Mapa de calor de criterios prioritarios**

Fuente: Elaboración propia

El mapa de calor que presenta la valoración de criterios para la selección de lenguajes de programación, según los roles de Arquitecto de Software, Desarrollador y Líder de Proyecto, revela una serie de tendencias y prioridades que reflejan las diferentes perspectivas de cada rol.

En primer lugar, el soporte y comunidad es un criterio altamente valorado por todos los roles, especialmente por los Líderes de Proyecto, quienes le otorgan una puntuación de 4.82. Esto indica que contar con una comunidad activa y un soporte técnico robusto es fundamental para garantizar la sostenibilidad y la resolución eficiente de problemas en los proyectos.

Por otro lado, la popularidad del lenguaje es un aspecto que los Arquitectos de Software y Desarrolladores consideran muy importante, con puntuaciones de 5.00 y 4.86 respectivamente. Esto sugiere que prefieren lenguajes ampliamente adoptados y con una gran cantidad de recursos disponibles. Sin embargo, los Líderes de Proyecto le dan menor relevancia a este criterio, con una puntuación de 3.83, lo que podría deberse a que su enfoque está más orientado hacia aspectos estratégicos y menos hacia la adopción general del lenguaje

El rendimiento del lenguaje es otro criterio clave, especialmente para los Desarrolladores, quienes lo valoran con 4.82. Esto refleja su preocupación por la eficiencia y velocidad del lenguaje en la ejecución de tareas. Los Arquitectos de Software y Líderes de Proyecto también consideran importante este aspecto, pero en menor medida, con puntuaciones de 4.13 y 3.90 respectivamente. En cuanto al propósito del proyecto, tanto Arquitectos de Software como Desarrolladores le otorgan una importancia similar, con puntuaciones de 4.00 y 4.14, lo que indica que buscan alinear el lenguaje con los objetivos técnicos del proyecto. Los Líderes de Proyecto, por su parte, le dan menor importancia a este criterio, con una puntuación de 3.67, posiblemente porque su enfoque está más centrado en la gestión y menos en los detalles técnicos

En cuanto a la compatibilidad con otros sistemas, todos los roles le otorgan una alta importancia, con puntuaciones de 4.68 para Arquitectos de Software, 4.64 para Desarrolladores y 4.82 para Líderes de Proyecto. Esto indica que la integración con sistemas existentes es un aspecto clave en la selección de lenguajes. Por otro lado, las necesidades del cliente son un criterio que los Líderes de Proyecto valoran menos, con una puntuación de 2.43, lo que sugiere que su enfoque está más en la viabilidad técnica que en los requerimientos funcionales del cliente.

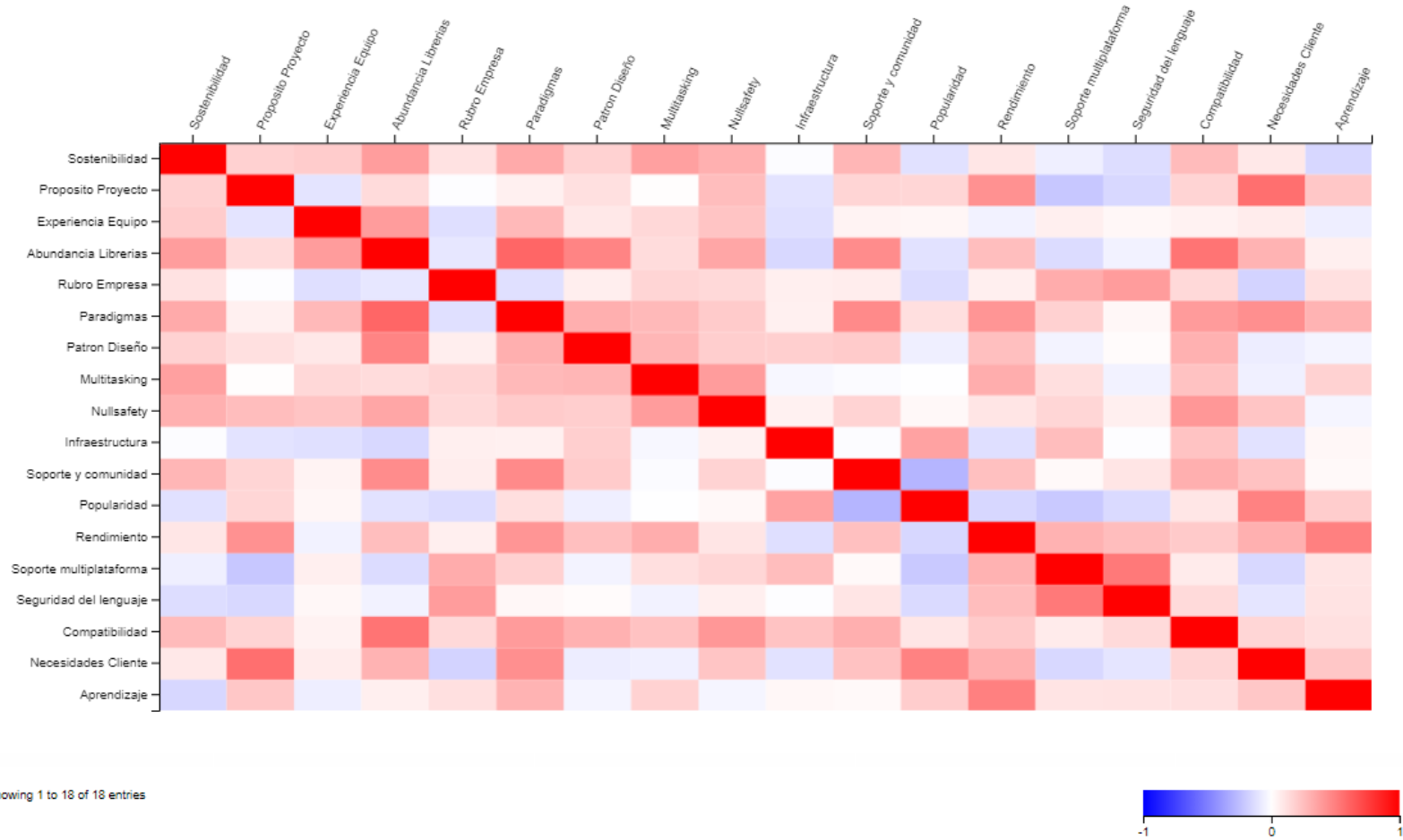
### **4.3 Matriz de Correlación**

Al igual que los demás análisis la matriz de correlación fue generada desde herramientas tecnológicas que facilitan el proceso y evitan la alteración de los resultados. Esta matriz es una herramienta fundamental para analizar la relación directa entre las variables.

Entre los objetivos de la matriz de correlación tenemos:

- Identificar relaciones positivas (mayor a 0).
- Identificar relaciones negativas (menor a 0).
- Identificar ausencia de relación (cercano a 0).

El resultado de la matriz de correlación generada puede verse en la siguiente ilustración.



**Ilustración 28** Mapa de calor matriz de correlación

Fuente: Elaboración propia.

#### 4.4 Análisis inferencial y modelos aplicados factores determinantes

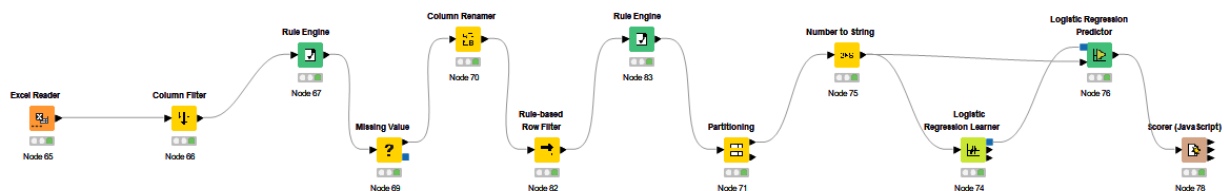
En la presente investigación se proponen los siguientes 3 modelos:

- Regresión Logística.
- Naive Bayes
- Random Forest.

Estos modelos fueron seleccionados debido a la naturaleza del presente estudio y los objetivos plasmados en el capítulo I. El uso de Regresión Logística, Naive Bayes y Random Forest permite abordar este análisis desde diferentes enfoques complementarios. Regresión Logística es útil para identificar qué características (certificaciones, metodologías ágiles, formación, etc.) tienen mayor impacto en la clasificación de un profesional en un determinado rol. Naive Bayes, al ser un modelo basado en probabilidades, permite evaluar la combinación de factores que más comúnmente se asocian con cada perfil. Random Forest, por su capacidad de manejar relaciones no lineales y reducir el sobreajuste, contribuye a identificar las características más relevantes en la clasificación de los roles.

##### 4.4.1 Regresión logística

La Regresión Logística es un modelo estadístico ampliamente utilizado para problemas de clasificación binaria y multiclase, lo que permite analizar qué factores influyen en la selección de un lenguaje de programación por parte de distintos roles (Desarrollador, Líder, Arquitecto). Para este modelo fue necesario realizar el siguiente diagrama en Knime.



#### Ilustración 29 Estructura regresión logística

Fuente: Elaboración propia

Los resultados del análisis de regresión logística se pueden apreciar en la siguiente tabla.

**Tabla 19 Regresión logística**

Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	30	1	96.77%
1 (Actual)	1	10	90.91%
	96.77%	90.91%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
0	30	1	10	1	96.77%	96.77%	96.77%	90.91%	96.77%
1	10	1	30	1	90.91%	90.91%	90.91%	96.77%	90.91%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa ( $\kappa$ )	Correctly Classified	Incorrectly Classified
95.24%	4.76%	0.877	40	2

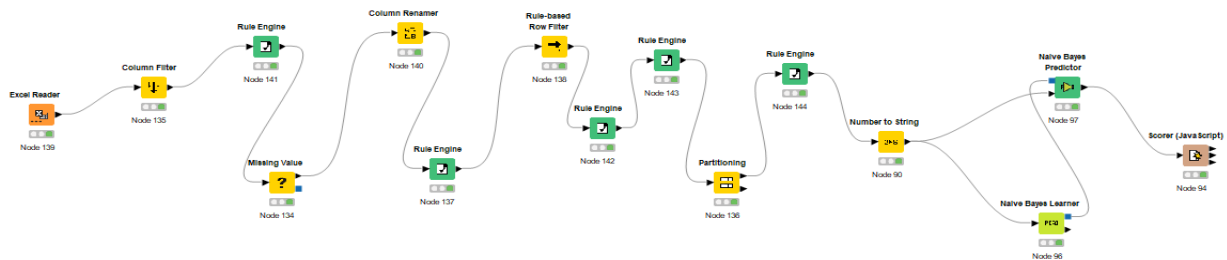
Fuente: Elaboración propia

Los resultados de este modelo reflejan un alto rendimiento con una exactitud del 95.24% y un Kapp de Cohen de 0.877 lo cual indica una fuerte concordancia entre las predicciones y valores reales.

Sobre la comparación de ambas clases los resultados muestran altos valores de recall, precisión y f-measure, lo cual sugiere que el modelo no está sesgado a una clase particular. En cuanto a errores, se obtiene únicamente 2 errores de clasificación lo que refuerza la robustez del modelo.

### 4.4.2 Naive Bayes

El modelo Naive Bayes es ideal para problemas de clasificación basada en probabilidades, especialmente cuando se trabaja con datos categóricos como preferencias tecnológicas y metodologías utilizadas. La construcción de este modelo en Knime fue realizada con el siguiente diagrama.



**Ilustración 30 Estructura Naive Bayes**

Fuente: Elaboración propia

El modelo Naive Bayes otorgo los resultados que pueden verse en la siguiente tabla.

**Tabla 20 Naive Bayes**

Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	28	3	90.32%
1 (Actual)	1	10	90.91%
	96.55%	76.92%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
0	28	1	10	3	90.32%	96.55%	90.32%	90.91%	93.33%
1	10	3	28	1	90.91%	76.92%	90.91%	90.32%	83.33%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa ( $\kappa$ )	Correctly Classified	Incorrectly Classified
90.48%	9.52%	0.767	38	4

Fuente: Elaboración propia

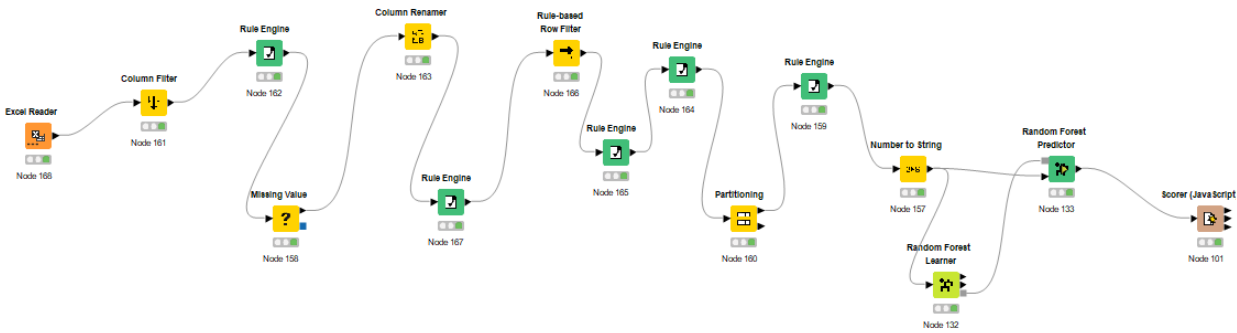
El modelo Naive Bayes muestra un desempeño aceptable, con una precisión del 90.48% y un coeficiente Kappa de Cohen de 0.767, lo que indica una concordancia moderada entre las predicciones y los valores reales.

Al analizar ambas clases, se observa que el modelo mantiene un equilibrio entre las métricas de recall, precisión y F-measure, lo que sugiere que no favorece excesivamente a una clase sobre la otra. Aunque se produjeron 4 errores de clasificación, el modelo sigue siendo confiable en términos generales, aunque aún tiene margen para mejorar su exactitud.

#### 4.4.3 Random Forest

El siguiente modelo para evaluar será el de Random Forest. Este modelo fue introducido en el 2001 y es una técnica una técnica de aprendizaje supervisado que se basa en la construcción de múltiples árboles de decisión durante el entrenamiento. La idea central detrás de este método es combinar las predicciones de varios árboles para obtener un modelo más robusto y generalizable, reduciendo así el riesgo de sobreajuste que puede ocurrir con un solo árbol de decisión (Espinosa Zúñiga, 2020).

El esquema utilizado dentro de Knime para realizar este modelo puede verse a continuación.



### Ilustración 31 Estructura Random Forest

Fuente: Elaboración propia

Los resultados obtenidos serán mostrados en la siguiente tabla.

**Tabla 21 Random Forest**

Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	31	1	96.88%
1 (Actual)	1	9	90.00%
	96.88%	90.00%	

Class Statistics

Class	True Positives	False Positives	True Negatives	False Negatives	Recall	Precision	Sensitivity	Specificity	F-measure
0	31	1	9	1	96.88%	96.88%	96.88%	90.00%	96.88%
1	9	1	31	1	90.00%	90.00%	90.00%	96.88%	90.00%

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa ( $\kappa$ )	Correctly Classified	Incorrectly Classified
95.24%	4.76%	0.869	40	2

Fuente: Elaboración propia

El modelo Random Forest muestra un rendimiento sólido, alcanzando una precisión del 95.24% y un coeficiente Kappa de Cohen de 0.869, lo que indica una alta concordancia entre las predicciones y los valores reales.

En la evaluación de ambas clases, se observa que el modelo mantiene un equilibrio en las métricas de recall, precisión y F-measure, lo que sugiere que clasifica de manera consistente sin favorecer a una clase en particular. Además, el modelo cometió solo 2 errores de clasificación en todo el conjunto de datos evaluado. Este bajo número de errores refuerza la idea de que el modelo es altamente confiable y efectivo. La exactitud general del modelo es notablemente alta, lo que sugiere que sus predicciones son precisas y consistentes en la mayoría de los casos.

Este desempeño excepcional no solo demuestra la robustez del modelo, sino también su capacidad para generalizar correctamente a nuevos datos. La combinación de un alto

rendimiento en las métricas clave y un bajo número de errores de clasificación lo convierte en una herramienta valiosa para aplicaciones prácticas, donde la precisión y la consistencia son críticas.

Los datos resumidos se muestran a continuación.

**Tabla 22 Resumen modelos aplicados**

Estadísticas	Regresión logística	Naive Bayes	Random Forest
Recall	96.77%	90.32%	96.88%
Precision	96.77%	96.55%	96.88%
Sensitivity	96.77%	90.32%	96.88%
Specifity	90.91%	90.91%	90%
F-Measure	96.77%	93.33%	96.88%
Overall Accuracy	95.24%	90.48%	95.24%
Overall Error	4.76%	9.52%	4.76%

Fuente: Elaboración propia.

Luego de ejecutar los tres modelos, se observó que los modelos de Regresión Logística y Random Forest presentaron un mayor grado de confiabilidad y robustez. La Regresión Logística destacó por su interpretabilidad y estabilidad, proporcionando coeficientes claros que permiten entender el impacto de cada factor en la elección de lenguajes de programación. Por otro lado, Random Forest demostró una capacidad superior para manejar la complejidad de los datos, capturando interacciones no lineales y proporcionando métricas de importancia de características. Ambos modelos mostraron un rendimiento equilibrado en términos de precisión, recall y f-measure, con un número mínimo de errores de clasificación.

## CAPITULO V – CONCLUSIONES Y RECOMENDACIONES

### 5.1 Conclusiones

- Los distintos roles seleccionados tienen una preferencia marcada por ciertos aspectos claves como lo son la abundancia de librerías, la compatibilidad con otros sistemas y la popularidad del lenguaje con una media de 4.72 cada uno, no muy lejos queda la importancia que se le da al rendimiento de lenguaje y la facilidad de aprendizaje con una media de 4.69 y 4.67 respectivamente.
- Se observó un comportamiento distinto de aspectos a incluir según el rol profesional, por ejemplo, Los arquitectos de software muestran una clara inclinación hacia la compatibilidad con otros sistemas, lo que indica su preocupación por la interoperabilidad y la integración tecnológica. Además, valoran la abundancia de librerías y la facilidad de aprendizaje, sugiriendo que buscan herramientas con un ecosistema sólido y una curva de aprendizaje manejable. Sin embargo, otorgan menor importancia a las necesidades del cliente, lo que sugiere que su enfoque está más orientado a la infraestructura técnica que a los requerimientos funcionales. Esto sugiere que los criterios a incluir en el instrumento deben tener un enfoque flexible y adaptable que no limite según el rol profesional de la persona.

### 5.2 Recomendaciones

- Dado que los distintos roles priorizan aspectos como la abundancia de librerías, la compatibilidad con otros sistemas y la popularidad del lenguaje es recomendable que la selección de un lenguaje de programación en un equipo de desarrollo considere estos factores como pilares fundamentales. Esto implica optar por lenguajes con un ecosistema sólido, amplia integración con otras tecnologías y un alto nivel de adopción en la industria, ya que estos elementos facilitan el desarrollo, el mantenimiento y la escalabilidad de los proyectos. Por otro lado, el hecho de que factores como el rubro de la empresa, la infraestructura, la sostenibilidad y la experiencia del equipo tengan valores considerablemente bajos sugiere una oportunidad de mejora en la toma de decisiones. Aunque estos aspectos no son los más influyentes en la elección de un lenguaje, su impacto a largo plazo podría ser significativo. Se recomienda no

descartarlos por completo, sino integrarlos en la evaluación para garantizar que el lenguaje seleccionado sea viable en el contexto específico de la organización, sostenible en el tiempo y adecuado al nivel de experiencia del equipo, evitando así posibles dificultades en la implementación y mantenimiento del software.

- Dado que los diferentes roles profesionales priorizan aspectos distintos en la selección de lenguajes de programación, y que incluso dentro de un mismo rol no siempre hay consenso sobre los factores más importantes, se recomienda diseñar un instrumento flexible que integre los elementos prioritarios identificados en el análisis de datos. Este instrumento debe incluir tanto factores técnicos como factores estratégicos y de gestión, inspirados en marcos de referencia como ITIL. La inclusión de estos elementos no solo asegurará que el instrumento sea útil para todos los roles involucrados, sino que también permitirá alinear la selección de lenguajes de programación con las mejores prácticas de gestión de servicios de TI, garantizando que las decisiones estén alineadas con los objetivos técnicos, funcionales y estratégicos de la organización

## **CAPITULO VI – APLICABILIDAD**

### **6.1 Instrumento para la elección de lenguajes de programación**

Este título refleja un enfoque estructurado y analítico para la selección de lenguajes de programación, considerando múltiples dimensiones clave como el propósito del desarrollo, el rendimiento, la escalabilidad, la compatibilidad con tecnologías existentes y las tendencias del mercado. El "Instrumento para la Elección de Lenguajes de Programación" enfatiza la necesidad de un proceso bien informado y contextualizado para la toma de decisiones en diversos entornos de desarrollo.

Este instrumento se construye sobre una base sólida de investigación y análisis de datos, proporcionando criterios claros y metodologías efectivas para evaluar las opciones de lenguajes según distintos roles profesionales, necesidades empresariales y tendencias tecnológicas emergentes. Su enfoque estratégico permitirá a los distintos roles tomar decisiones fundamentadas que optimicen la eficiencia y sostenibilidad de sus proyectos.

Además, tomando en cuenta que las decisiones rara vez se toman de forma individual, esta medida fomenta la colaboración y la participación. El instrumento está diseñado para propiciar que cada miembro contribuya con sus opiniones, lo que enriquece el proceso de toma de decisiones y asegura que se consideren todos los aspectos relevantes.

### **6.2 Justificación de la propuesta**

El mundo tecnológico se encuentra en constante evolución, la elección del lenguaje de programación adecuado es un factor determinante para el éxito de los proyectos de software. Sin embargo, esta decisión no siempre es sencilla, ya que depende de múltiples variables, como el propósito del desarrollo, el rendimiento requerido, la escalabilidad, la compatibilidad con tecnologías existentes y la curva de aprendizaje.

Actualmente, la falta de un criterio unificado para seleccionar un lenguaje de programación puede llevar a decisiones basadas en preferencias personales o tendencias del mercado, sin considerar factores estratégicos que optimicen el desarrollo y mantenimiento del software a largo plazo. Esto puede derivar en costos innecesarios, problemas de integración y dificultades en la adopción de nuevas tecnologías. Por esta razón, es fundamental contar con un instrumento estructurado que facilite la elección de lenguajes de programación.

### **6.3 Alcance de la propuesta**

Desarrollar un instrumento que facilite la elección de lenguajes de programación para equipos de desarrollo en empresas, basado en factores técnicos, estratégicos y contextuales.

#### **Objetivos**

- Diseñar una herramienta flexible que permita evaluar y comparar lenguajes de programación según los factores identificados que facilite la toma de decisiones.
- Integrar principios y prácticas de ITIL en el diseño de la herramienta, asegurando que los criterios de selección de lenguajes de programación consideren aspectos clave de la gestión de servicios de TI, como la disponibilidad, la sostenibilidad, el rendimiento, la mantenibilidad y la mejora continua.

#### **Exclusiones**

- El presente instrumento puede ser aplicado para cualquier tipo de proyecto de desarrollo de software, no se limita a un enfoque específico como puede ser el desarrollo web, de dispositivos móviles y aplicaciones de escritorio.
- La herramienta puede ser utilizada desde encargados de proyecto hasta programadores juniors, claro está que se orienta a roles de dirección que cuenten con un panorama más amplio de la empresa o contexto donde se implementará el proyecto.

### **6.4 Descripción y desarrollo**

En esta sección se describe la propuesta para optimizar la selección de lenguajes de programación en el contexto de desarrollo de software, con el fin de mejorar la eficiencia y calidad de los proyectos. La propuesta se centra en diseñar un instrumento (herramienta o metodología) que permita evaluar y comparar lenguajes de programación según factores clave identificados, considerando las perspectivas de diferentes roles profesionales (desarrolladores, arquitectos de software y líderes de proyecto).

El instrumento integrará principios y prácticas de ITIL para asegurar que los criterios de selección estén alineados con las mejores prácticas de gestión de servicios de TI, como la disponibilidad, la capacidad y la mejora continua. Además, se fomentará la colaboración entre roles para garantizar que las decisiones sean consensuadas y equilibradas, tomando en cuenta tanto aspectos técnicos como estratégicos.

### 6.4.1 Aplicación práctica a la investigación

Los fundamentos para la optimización de la elección de lenguajes de programación se adquieren a través del análisis y estudio de los datos recopilados. Los resultados indican aspectos que sobresalen por los demás, como la compatibilidad con sistemas existentes, la escalabilidad, la facilidad de aprendizaje y la alineación con las tendencias tecnológicas emergentes. Estos factores son críticos para garantizar que las decisiones de selección de lenguajes estén alineadas con los objetivos técnicos y estratégicos de los proyectos.

### 6.4.2 Diseño del instrumento

La herramienta sugerida incluye aspectos relevantes identificados en la investigación. Esta herramienta no solo quedará plasmada en el documento como un instrumento útil, sino que también tiene el potencial de ser implementada en un lenguaje de programación, lo que permitiría su uso en formato digital y evitaría el gasto innecesario de papel.

Al desarrollar la herramienta en un formato digital (por ejemplo, una aplicación web o una hoja de cálculo interactiva), se contribuye a las medidas de sostenibilidad ambiental. Además, una versión digital de esta propuesta ofrece ventajas adicionales como la accesibilidad de este y puede formar parte de la documentación de un proyecto de software que sirva a forma de gestionar el conocimiento y entender los factores y motivos que llevaron a elegir un lenguaje de programación específicamente.

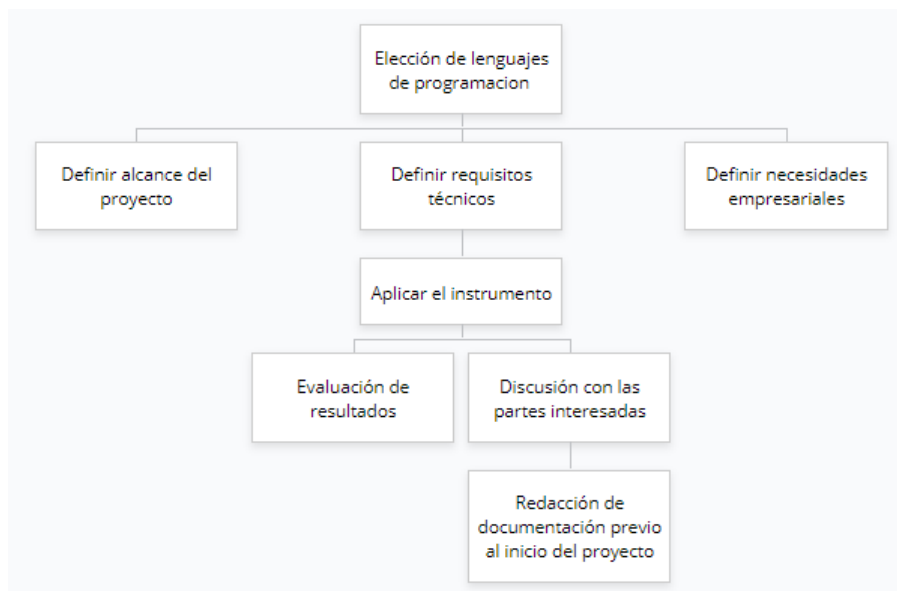
**Tabla 23 Instrumento para la elección de lenguajes de programación**

Instrumento que facilite la selección de lenguajes de programación						
<b>Instrucciones de uso.</b>						
<b>Asignar pesos:</b> Cada factor tiene un peso predeterminado, siendo 1 el más bajo y 3 el más alto.						
<b>Evaluar lenguajes:</b> Para cada lenguaje de programación, asigna una puntuación (1 a 3) en cada factor.						
<b>Calcular puntuación total:</b> Sumar los resultados para obtener la puntuación total de cada lenguaje.						
<b>Comparar resultados:</b> El lenguaje con la puntuación más alta es la mejor opción según los criterios evaluados.						
<b>Nombre del proyecto:</b>				<b>Importancia del proyecto:</b>		
Criterio	Bajo (1)	Medio (2)	Alto (3)	Lenguaje 1	Lenguaje 2	Lenguaje 3
Facilidad Aprendizaje	Complejo	Moderado	Sencillo			
Rendimiento	Lento	Aceptable	Rápido			
Soporte	Escaso	Moderado	Amplio			
Compatibilidad	Limitado	Algunas opciones	Amplia gama			

Criterio	Bajo (1)	Medio (2)	Alta (3)	Lenguaje 1	Lenguaje 2	Lenguaje 3
Costos Asociados	Altos	Moderados	Bajos			
Popularidad	Baja	Moderado	Alta			
Escalabilidad	Baja	Moderada	Alta			
Seguridad	Baja	Moderada	Alta			
Disponibilidad (ITIL)	Bajo	Moderada	Alta			
Mejora continua (ITIL)	Sin soporte	Básico	Avanzado			
Mantenibilidad	Bajo	Moderada	Alta			
Multiplataforma	Incompatible	Media	Compatible			
Abundancia de librerías	Baja	Media	Alta			
Experiencia del equipo con el lenguaje	Baja	Moderada	Alta			
Proyectos en el mismo lenguaje	Ninguno o casi inexistentes	Medio	Muchos			
Valoración final						

Fuente: Elaboración propia

### 6.4.3 Flujo de proceso de selección de lenguajes



**Ilustración 32** Flujo de proceso de selección de lenguajes

Fuente: Elaboración propia

### 6.4.3 Guía de uso del instrumento

A continuación, se presenta la guía de uso propuesta para utilizar el instrumento en una empresa:

- **Levantamiento de necesidades en la empresa:** Para hacer uso efectivo del instrumento, se deben identificar los requerimientos técnicos y las condiciones contextuales específicas de la empresa. Se recomienda realizar un proceso de consultoría interna dentro del departamento de TI involucrando a los distintos roles que participan en el desarrollo de software. Este levantamiento puede llevarse a cabo mediante entrevistas, encuestas, revisiones de documentación de proyectos anteriores o análisis de políticas y regulaciones internas vigentes.
- **Seleccionar lenguajes a comparar:** Una vez identificadas las necesidades, se deben definir los lenguajes de programación candidatos a ser evaluados en función de su viabilidad dentro del contexto identificado. El contexto identificado debe abarcar desde requisitos técnicos y factores internos de la empresa.
- **Aplicación del instrumento:** Con los lenguajes definidos, se procede a aplicar el instrumento, asignando valores del 1 al 3 para cada criterio, donde 1 representa un nivel bajo y 3 un nivel alto.
- **Evaluación de resultados:** Al tener el instrumento completo se debe revisar el lenguaje con mayor valoración y si este cumple con los requisitos técnicos y contextuales de cada empresa se selecciona.

Los pasos a seguir para su implementación se muestra en el flujograma 6.6.2.

### 6.5 Medidas de control

Para evaluar la efectividad de la rúbrica en la selección de lenguajes de programación, se utilizarán medidas basados en la precisión de la elección, la eficiencia en la toma de decisiones y la satisfacción de los usuarios que implementen la herramienta. Estas medidas permitirán medir el impacto del instrumento en la optimización del proceso de selección de tecnologías.

Las medidas de control propuestas son las siguientes:

- **Precisión en la elección del lenguaje:** Se medirá la cantidad de ajustes o reconsideraciones que se deben hacer después de haber seleccionado un lenguaje con la rúbrica. Un menor

número de cambios indicará una mayor efectividad del modelo de selección.

- **Tiempo de decisión:** Se evaluará el tiempo que tardan los encargados de desarrollo en elegir un lenguaje de programación utilizando la rúbrica, comparándolo con decisiones previas sin esta herramienta. Un menor tiempo de selección será indicativo de una mejor eficiencia en la toma de decisiones.
- **Satisfacción del usuario:** Se recopilarán opiniones de los encargados de desarrollo sobre la utilidad de la herramienta, utilizando encuestas o entrevistas. Se medirán factores como la facilidad de uso, la claridad de los criterios y la percepción de mejora en la selección del lenguaje.
- **Alineación con los requisitos del proyecto:** Se analizará si el lenguaje elegido con la rúbrica se ajusta a las necesidades del negocio y de los equipos de desarrollo, comparando las características priorizadas con los objetivos del proyecto.

### 6.5.1 Plan de Seguimiento

**Tabla 24 Plan de seguimiento**

Fase	Responsable(s)	Tareas
<b>1. Definición de Objetivos y Criterios</b>	Líder del proyecto / Arquitecto de software	- Validar criterios de selección.
		- Ajustar los pesos de los factores.
		- Obtener aprobación de stakeholders.
<b>2. Recopilación de Datos</b>	Analistas de software / Desarrolladores senior	- Evaluar lenguajes en cada criterio.
		- Consultar documentación y benchmarks.
		- Recabar opiniones de expertos.
<b>3. Aplicación del Instrumento y Cálculo de Puntuaciones</b>	Equipo de arquitectura / Líder técnico	- Asignar puntuaciones a cada lenguaje.
		- Calcular la puntuación total.

Fase	Responsable(s)	Tareas
		- Revisar inconsistencias en la evaluación.
<b>4. Análisis de Resultados y Selección del Lenguaje</b>	Comité de evaluación (Líder del proyecto, Arquitecto, CTO, Desarrolladores senior)	- Comparar puntuaciones obtenidas.
		- Evaluar si el lenguaje cumple con los objetivos del proyecto.
		- Documentar la justificación de la elección.
<b>5. Validación con Stakeholders y Ajustes Finales</b>	Líder del proyecto / Equipo de desarrollo	- Presentar resultados a las partes interesadas.
		- Recibir retroalimentación y hacer ajustes.
		- Confirmar la selección del lenguaje.
<b>6. Seguimiento y Monitoreo de la Implementación</b>	PMO / Arquitecto / Líder de desarrollo	- Evaluar desempeño del lenguaje en la implementación real.
		- Identificar problemas y optimizaciones.
		- Realizar ajustes si es necesario.

Fuente: Elaboración propia

## 6.6 Cronograma de implementación y presupuesto

Para el cronograma y el presupuesto se utilizará el método de Estimación por Tres Valores (PERT) el cual es utilizado para la gestión de proyectos con incertidumbre. Entre los beneficios de este modelo tenemos (Project Management Institute, 2015):

- Mejora la planificación y programación de actividades.
- Previsión mejorada de las necesidades de recursos.
- Identificación de patrones de planificación repetitivos que pueden seguirse en otros proyectos, simplificando así el proceso de planificación.

- Estimación basada en fórmula matemática.
- Seguimiento y control efectivo facilitando la comparación entre lo planeado y el progreso actual del proyecto.
- Capacidad de ver y, por tanto, reprogramar actividades para reflejar las dependencias entre proyectos y las limitaciones de recursos siguiendo reglas de prioridad conocidas.
- Sugiere el tiempo esperado de finalización del proyecto, la probabilidad de finalización antes de una fecha específica, las actividades de la ruta crítica que impactan el tiempo de finalización, las actividades que tienen tiempo de holgura y que pueden prestar recursos a las actividades de la ruta crítica y las fechas de inicio y finalización de las actividades.

Para la distribución triangular se utiliza la siguiente formula:

$$tE: (tO + tM + tP) / 3$$

Para la distribución beta se utiliza la siguiente formula:

$$tE: (tO + 4tM + tP) / 6$$

### 6.6.1 Cronograma de implementación en semanas

**Tabla 25 Cronograma de implementación en semanas**

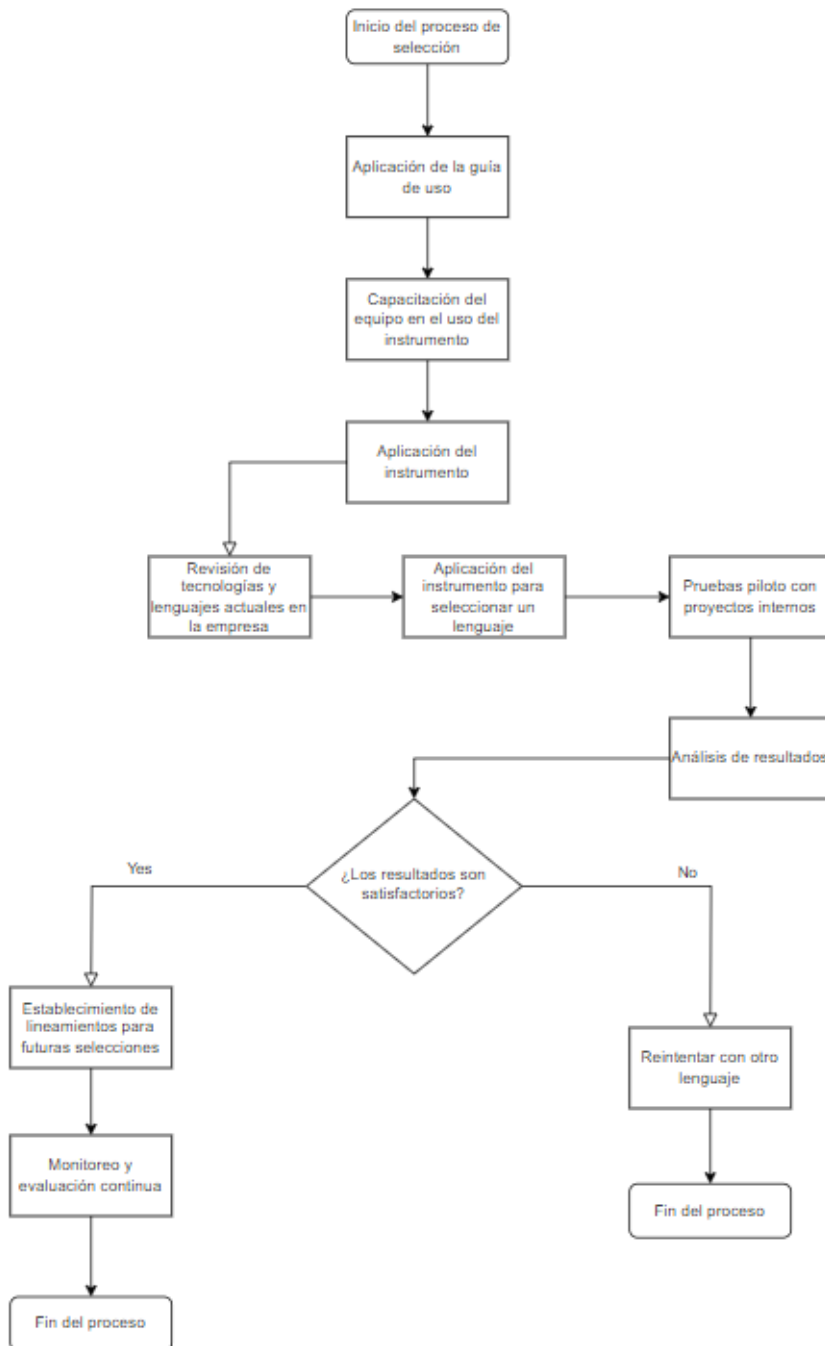
Actividad	Optimista (O)	Mas probable (M)	Pesimista (P)	Triangular	Beta
Adaptar el instrumento según necesidades de la empresa	2	3	4	3	3
Capacitación del equipo en el uso del instrumento	1	2	3	2	2
Revisión de tecnologías y lenguajes actuales en la empresa	2	3	4	3	3
Aplicación del instrumento en la selección de un nuevo lenguaje	2	4	6	4	4
Evaluación de lenguajes con el instrumento	3	5	7	5	5

Actividad	Optimista (O)	Mas probable (M)	Pesimista (P)	Triangular	Beta
Pruebas piloto con proyectos internos	4	6	8	6	6
Análisis de impacto y ajustes según resultados	3	5	7	5	5
Establecimiento de lineamientos y procesos para futuras selecciones	2	4	6	4	4
Monitoreo y evaluación continua del proceso de selección	3	5	7	5	5
Reserva de contingencias para actividades (Total de Actividades)				8	6
Reserva de Contingencias (Total de paquetes de trabajo)				10	8
Línea Base de Costos (BAC)				80	90
Reserva de Gestión				5	3
Total				137	141

Fuente: Elaboración propia

## 6.6.2 Flujo de implementación del proceso

En la siguiente ilustración se presenta el flujo de implementación del instrumento.



**Ilustración 33** Flujo de implementación del proceso  
Fuente: Elaboración propia

### 6.6.3 Presupuesto del proyecto en Lempiras

**Tabla 26 Presupuesto en Lempiras**

Actividad	Optimista (O)	Mas probable (M)	Pesimista (P)	Triangular	Beta
Capacitación del equipo en el uso del instrumento	1,000	2,000	3,000	3,500	5,000
Revisión de tecnologías y lenguajes actuales en la empresa	2,000	3,000	4,000	5,000	7,500
Aplicación del instrumento en la selección de un nuevo lenguaje	5,000	7,000	9,000	11,000	15,000
Evaluación de lenguajes con el instrumento	6,000	8,000	10,000	12,500	17,000
Pruebas piloto con proyectos internos	8,000	10,000	12,000	14,500	19,000
Análisis de impacto y ajustes según resultados	5,000	7,000	9,000	11,000	15,000
Establecimiento de lineamientos y procesos para futuras selecciones	4,000	6,000	8,000	10,000	13,500
Monitoreo y evaluación continua del proceso de selección	5,000	7,000	9,000	11,500	16,000
Reserva de contingencias para actividades (Total de Actividades)				14,000	18,000
Reserva de Contingencias (Total de paquetes de trabajo)				15,000	20,000
Línea Base de Costos (BAC)				120,000	150,000
Reserva de Gestión				11,000	15,000
Total				183,000	224,000

Fuente: Elaboración propia

## 6.7 Matriz de Concordancia

**Tabla 27 Matriz de Concordancia**

Capítulo I			Capítulo II	Capítulo III			Capítulo V	Capítulo VI	
Título investigación	Objetivo General	Objetivos Específicos	Teorías de sustento / Metodologías	Variables	Muestra	Técnicas	Conclusiones	Propuesta	Objetivos propuesta
Factores influyentes en la selección de lenguajes de programación en el desarrollo web en honduras	Diagnosticar las preferencias de los profesionales del software en el contexto del desarrollo web para comprender los criterios que guían la elección de lenguajes de programación, con el propósito de identificar	1. Identificar los 5 principales factores que influyen en la selección de un lenguaje de programación entre profesionales del área de desarrollo de software. 2. Diseñar una rúbrica compuesta de los factores identificados que sirva como herramienta de orientación para los responsables de desarrollo	-Cuarta Revolución Industrial. - ITIL. - Marco de Gestión de Proyectos Ágil Scrum.	Rol Profesional, Conocimiento de ITIL, Aceptación de rúbrica, Años de experiencia, Metodologías en cascada, Metodologías ágiles, Factores considerados relevantes,  Propósito de proyecto, sostenibilidad y mantenibilidad, paquetes, rubro de la empresa, factores técnicos del lenguaje, ITIL, aspectos de ITIL	Muestreo probabilístico. Clasificando los participantes en roles relevantes del rubro del desarrollo de software.	Estratificación	1. Los distintos roles seleccionados tienen una preferencia marcada por ciertos aspectos claves como lo son la abundancia de librerías, la compatibilidad con otros sistemas y la popularidad del lenguaje con una media de 4.72 cada uno, no muy lejos queda la importancia que se le da al rendimiento de lenguaje y la facilidad de aprendizaje con una media de 4.69 y 4.67 respectivamente. En cambio, los valores con	Instrumento para la elección de lenguajes de programación	Diseñar una herramienta flexible que permita evaluar y comparar lenguajes de programación según los factores identificados que facilite la toma de decisiones.  Integrar principios y prácticas de ITIL en el diseño de la herramienta, asegurando que los criterios de selección de lenguajes de programación consideren aspectos clave de la gestión de servicios de TI, como la disponibilidad

	factores determinante s en dicha selección.	en empresas al momento de seleccionar un lenguaje de programación.				<p>menos media son el rubro de la empresa, la infraestructura, la sostenibilidad y la experiencia del equipo, con valores sumamente bajos que no llegan ni al 0.90 de media.</p> <p>2. Se observó un comportamiento distinto de aspectos a incluir según el rol profesional, por ejemplo, Los arquitectos de software muestran una clara inclinación hacia la compatibilidad con otros sistemas, lo que indica su preocupación por la interoperabilidad y la integración tecnológica. Además, valoran la</p>	d, la sostenibilidad, el rendimiento, la mantenibilidad y la mejora continua.
--	---	--	--	--	--	--	---

							<p>abundancia de librerías y la facilidad de aprendizaje, sugiriendo que buscan herramientas con un ecosistema sólido y una curva de aprendizaje manejable. Sin embargo, otorgan menor importancia a las necesidades del cliente, lo que sugiere que su enfoque está más orientado a la infraestructura técnica que a los requerimientos funcionales. Esto sugiere que los criterios deben tener un enfoque flexible y adaptable que no limite según el rol profesional de la persona.</p>		
--	--	--	--	--	--	--	--	--	--

Fuente: Elaboración propia

## REFERENCIAS BIBLIOGRÁFICAS

Agency, M. D. (2021, enero 4). ¿Cuántos lenguajes de programación existen? *Epitech Spain*.

<https://www.epitech-it.es/cuantos-lenguajes-existen/>

Amazon Web Services. (2024). *Precios de las instancias bajo demanda de EC2 – Amazon Web*

*Services*. Amazon Web Services, Inc. <https://aws.amazon.com/es/ec2/pricing/on-demand/>

*Arquitectura de Software: Conceptos y ciclo de desarrollo* (Primera edición) (with Cervantes

Maceda, H., Castro Careaga, L., & Velasco-Elizondo, P.). (2016). Cengage Learning.

Atlassian. (2024). *Comparación entre la arquitectura monolítica y la arquitectura de*

*microservicios*. Atlassian. <https://www.atlassian.com/es/microservices/microservices-architecture/microservices-vs-monolith>

AXELOS. (2019). *ITIL foundation: ITIL 4 edition* (First edition). TSO (The Stationery Office).

Banco Central de Costa Rica. (2022). *BCCR - Inicio*. [https://www.bccr.fi.cr/indicadores-](https://www.bccr.fi.cr/indicadores-economicos/DocCuentasNacionales2017/CR_exportacion_servicios_redes_tic.pdf)

[economicos/DocCuentasNacionales2017/CR\\_exportacion\\_servicios\\_redes\\_tic.pdf](https://www.bccr.fi.cr/indicadores-economicos/DocCuentasNacionales2017/CR_exportacion_servicios_redes_tic.pdf)

Banco Mundial. (2024). *Guatemala Panorama general* [Text/HTML]. World Bank.

<https://www.bancomundial.org/es/country/guatemala/overview>

Cascadas, P. (2011). *GUÍA PARA LA ELECCIÓN DEL LENGUAJE DE DESARROLLO PARA PROYECTOS*.

CEPAL, O. (2021). *Tecnologías digitales para un nuevo futuro*.

DATA.AI. (2024). *Estado de móvil en 2024: El mercado de las aplicaciones se recuperó en*

*2023 y se prepara para un 2024 récord* | *data.ai Blog*. data.ai.

<https://www.data.ai/es/insights/market-data/state-of-mobile-2024/>

Espinosa Zúñiga, J. J. (2020). Aplicación de algoritmos Random Forest y XGBoost en una base

de solicitudes de tarjetas de crédito. *Ingeniería Investigación y Tecnología*, 21(3), 1-16.

<https://doi.org/10.22201/fi.25940732e.2020.21.3.022>

Esteban, M. D. (2022). *DESARROLLO DE LAS TIC'S EN EL TURISMO COMUNITARIO DE MÉXICO: Caso de Estudio Oaxaca*.

Folgueiras Bertomeu, P. (2016). *La entrevista*.

<https://diposit.ub.edu/dspace/bitstream/2445/99003/1/entrevista%20pf.pdf>

Gallego, V. (2023). *ANÁLISIS DE RENDIMIENTO DE LENGUAJES DE DESARROLLO DE APLICACIONES Y SERVICIOS WEB* [Universitat Politècnica de Catalunya].

<https://upcommons.upc.edu/bitstream/handle/2117/407069/181292.pdf?sequence=2&isAllowed=y>

*GDPR UE*. (2016).

Google Cloud. (2024). *¿Qué es la inteligencia artificial o IA?* Google Cloud.

<https://cloud.google.com/learn/what-is-artificial-intelligence>

HEARTBLEED: A CASE STUDY. (2018). *Issues In Information Systems*.

[https://doi.org/10.48009/2\\_iis\\_2018\\_99-108](https://doi.org/10.48009/2_iis_2018_99-108)

Hema, V., Thota, S., Naresh Kumar, S., Padmaja, C., Rama Krishna, C. B., & Mahender, K.

(2020). Scrum: An Effective Software Development Agile Tool. *IOP Conference Series: Materials Science and Engineering*, 981(2), 022060. <https://doi.org/10.1088/1757-899X/981/2/022060>

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). *Metodología de la investigación* (5a ed). McGraw-Hill.

Hernández Sampieri, R., & Fernandez-Collado, C. F. (2014). *Metodología de la investigación* (P. Baptista Lucio, Ed.; Sexta edición). McGraw-Hill Education.

Hinojosa Mamani, J., Mamani Gamarra, J. E., Jilaja Carita, E. E., Albarracín Machicado, F. D.,

- & Zela Paricahua, M. (2023). Infraestructura tecnológica y aprendizaje por competencias en la educación superior universitaria, Puno – Perú: Technological infrastructure and competency-based learning in university higher education, Puno – Peru. *LATAM Revista Latinoamericana de Ciencias Sociales y Humanidades*, 4(2).  
<https://doi.org/10.56712/latam.v4i2.986>
- IBM. (2021, septiembre 23). *¿Qué es la biblioteca de infraestructura de TI (ITIL)?* | IBM.  
<https://www.ibm.com/mx-es/topics/it-infrastructure-library>
- IEEE, De Russis, L., & Saenz, J. P. (2020). How is Open Source Software Development Different in Popular IoT Projects? *IEEE Access*, 8, 28337-28348.  
<https://doi.org/10.1109/ACCESS.2020.2972364>
- Índice Mundial de Innovación de 2024: Suiza, Suecia, Estados Unidos, Singapur y Reino Unido encabezan la clasificación; China, Türkiye, la India, Viet Nam y Filipinas, entre los países que más han subido en 10 años; nubarrones para las inversiones en innovación.* (2024). [https://www.wipo.int/pressroom/es/articles/2024/article\\_0013.html](https://www.wipo.int/pressroom/es/articles/2024/article_0013.html)
- INE. (2022, diciembre 17). *Cifras de población con acceso a tecnologías de información y comunicaciones, EPHPM junio 2022—INE*. <https://ine.gob.hn/v4/2022/12/17/cifras-de-poblacion-con-acceso-a-tecnologias-de-informacion-y-comunicaciones-junio-2022/>
- Intel. (2025). *Intel in Costa Rica*. Intel. <https://www.intel.com/content/www/xl/es/corporate-responsibility/intel-in-costa-rica.html>
- JetBrains*. (2021a). *JetBrains: Developer Tools for Professionals and Teams*.  
<https://www.jetbrains.com/lp/devecosystem-2021>
- JetBrains*. (2021b). *PHP Programming—The State of Developer Ecosystem in 2021 Infographic*.  
*JetBrains: Developer Tools for Professionals and Teams*.

- <https://www.jetbrains.com/idea/devecosystem-2021>
- Klaus Schwab. (2016). *La cuarta revolución industrial*. OverDrive.  
<https://www.overdrive.com/media/3014193/la-cuarta-revolucion-industrial>
- Lopez, R. (2020). *ANÁLISIS COMPARATIVO DE LENGUAJES DE PROGRAMACIÓN PARA EL DESARROLLO DE APLICACIONES EN CIENCIA DE DATOS (3)* [Tecnológico Nacional de México].  
[https://rinacional.tecnm.mx/bitstream/TecNM/4155/1/MC\\_Ricardo\\_Gudiel\\_Lopez\\_Perez\\_2020.pdf](https://rinacional.tecnm.mx/bitstream/TecNM/4155/1/MC_Ricardo_Gudiel_Lopez_Perez_2020.pdf)
- NOW, R. I. (2024, noviembre 5). *Los planes de inversión en IT de las empresas de Honduras para los próximos 12 meses*. IT NOW. <https://www.itnow.connectab2b.com/post/los-planes-de-inversion-en-it-de-las-empresas-de-honduras-para-los-proximos-12-meses>
- Otero. (2018). (PDF) ENFOQUES DE INVESTIGACIÓN. En *ENFOQUES DE INVESTIGACIÓN*.  
[https://www.researchgate.net/publication/326905435\\_ENFOQUES\\_DE\\_INVESTIGACION](https://www.researchgate.net/publication/326905435_ENFOQUES_DE_INVESTIGACION)
- Otto, C. (2017, agosto 23). *1.040 millones perdidos en un día: Cuando un ordenador arruina a una gran empresa*. *elconfidencial.com*. [https://www.elconfidencial.com/tecnologia/2017-08-23/inteligencia-artificial-knight-capital-quebra-delta-airlines-british-airways\\_1429756/](https://www.elconfidencial.com/tecnologia/2017-08-23/inteligencia-artificial-knight-capital-quebra-delta-airlines-british-airways_1429756/)
- Project Management Intitute. (2015). *ProjectManagement.com—PERT*.  
[https://www.projectmanagement.com/wikis/293291/pert#=\\_](https://www.projectmanagement.com/wikis/293291/pert#=_)
- React Community. (2024). *Conoce al equipo – React*. <https://es.react.dev/community/team>
- Rodriguez, A., & Florentino, L. (2020). *ANÁLISIS E IMPLEMENTACIÓN DE LA*

*PROGRAMACIÓN DE SOFTWARE COMO ASIGNATURA BÁSICA EN LAS ESCUELAS PÚBLICAS Y PRIVADAS A NIVEL DE EDUCACIÓN MEDIA, EN LA CIUDAD DE SANTO DOMINGO DURANTE EL PERIODO SEPTIEMBRE-DICIEMBRE 2020. CASO DE ESTUDIO: INSTITUTO TECNOLÓGICO FABIO AMABLE MOTA (43) [Universidad APEC].*

[https://bibliotecaunapec.blob.core.windows.net/tesis/TESIS\\_CI\\_ISC\\_08\\_2020\\_ET210264.pdf](https://bibliotecaunapec.blob.core.windows.net/tesis/TESIS_CI_ISC_08_2020_ET210264.pdf)

Rodríguez-Herrera, J., Amante-Orozco, A., Muñoz-Robles, C. A., Pimentel-López, J., & Martínez-Montoya, J. F. (2023). Importancia del big data satelital, lenguajes de programación, y NDVI en el seguimiento de cultivos. *Agro-Divulgación*, 3(3).  
<https://doi.org/10.54767/ad.v3i3.204>

*Scrum Guides*. (2020). <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf>

*The Top 25 Economies in the World*. (2024). Investopedia.  
<https://www.investopedia.com/insights/worlds-top-economies/>

*TIOBE Index*. (2024). TIOBE. <https://www.tiobe.com/tiobe-index/>

UNESCO. (2021). *China | 2021 Science Report*.  
<https://www.unesco.org/reports/science/2021/es/china>

Universidad de Galileo. (2024). *La Importancia de la Programación en la Actualidad: Un Pilar en la Revolución Industrial Digital | Universidad Galileo*.  
<https://www.galileo.edu/noticias/la-importancia-de-la-programacion-en-la-actualidad-un-pilar-en-la-revolucion-industrial-digital/>

Universidad ORT Uruguay. (2024). *Qué es programar: Guía esencial para iniciarse en la*

*programación—Universidad ORT Uruguay*. <https://fi.ort.edu.uy/blog/que-es-programar-y-para-que-sirve>

Urquía Moraleda, A. (2021). *Lenguajes de programación*. UNED - Universidad Nacional de Educación a Distancia.

Velásquez-Restrepo, S. M., Londoño-Gallego, J. A., López-Romero, C., & Vahos, J. D. (2018). Desarrollo de una plataforma web multimedial para la elaboración de proyectos bajo la metodología de marco lógico. *Lámpsakos*, 1(18), 12.  
<https://doi.org/10.21501/21454086.2601>

Virkus, R. (2016). *GUÍA A LA GALAXIA DE APLICACIONES MÓVILES* (Vol. 16).  
<https://blog.facialix.com/wp-content/uploads/2021/11/guia-a-la-galaxia-de-aplicaciones-moviles.pdf>

## ANEXOS

**Tabla 28 Instrumento Realizado**

Universidad Tecnológica Centroamericana Maestría en Gestión de Tecnologías de la Información Maestrante: Hector David Sánchez Mejía	
Estimado usuario, a continuación, se presentan una serie de preguntas que permitirán recolectar la información necesaria para el análisis del caso de estudio. El objetivo de la presente es conocer su opinión sobre la elección de un lenguaje de programación, su relación con ITIL y la influencia que puede tener en un proyecto.  Las respuestas brindadas se mantendrán confidenciales y no serán compartidas con ningún agente externo.  El tiempo estimado para llenar la encuesta es de 35 minutos.	
<b><u>Evaluación de aspectos generales</u></b>	
1. Rol Profesional: <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Desarrollador</li> <li>• <input type="checkbox"/> Líder de proyecto</li> <li>• <input type="checkbox"/> Arquitecto de software</li> </ul>	2. Años de experiencia profesional: <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Menos de 5 años</li> <li>• <input type="checkbox"/> 5-10 años</li> <li>• <input type="checkbox"/> Más de 10 años</li> </ul>
3. Años de experiencia en proyectos relacionados al desarrollo web: <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Menos de 3 años</li> <li>• <input type="checkbox"/> 3-5 años</li> <li>• <input type="checkbox"/> Más de 5 años</li> </ul>	4. Conocimiento de metodologías en cascada para la gestión de proyectos: <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Avanzado</li> <li>• <input type="checkbox"/> Intermedio</li> <li>• <input type="checkbox"/> Básico</li> </ul>

	<ul style="list-style-type: none"> <li>• <input type="checkbox"/> Ninguno</li> </ul>
<p>5. Conocimiento de metodologías ágiles para la gestión de proyectos:</p> <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Avanzado</li> <li>• <input type="checkbox"/> Intermedio</li> <li>• <input type="checkbox"/> Básico</li> <li>• <input type="checkbox"/> Ninguno</li> </ul>	<p>6. Conocimiento de ITIL:</p> <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Avanzado</li> <li>• <input type="checkbox"/> Intermedio</li> <li>• <input type="checkbox"/> Básico</li> <li>• <input type="checkbox"/> Ninguno</li> </ul>
<p><b><u>Factores para elegir lenguajes de programación:</u></b></p>	
<p>7. ¿Cuáles son los tres principales factores que considera al seleccionar un lenguaje de programación?</p> <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Rendimiento</li> <li>• <input type="checkbox"/> Popularidad</li> <li>• <input type="checkbox"/> Compatibilidad con otros sistemas</li> <li>• <input type="checkbox"/> Comunidad de soporte</li> <li>• <input type="checkbox"/> Facilidad de aprendizaje</li> <li>• <input type="checkbox"/> Seguridad del lenguaje</li> <li>• <input type="checkbox"/> Otro: _____</li> </ul>	<p>8. En una escala de 1 a 5, ¿qué tan importante es cada uno de los siguientes factores para elegir un lenguaje?</p> <ul style="list-style-type: none"> <li>• Facilidad de integración con otros sistemas _____</li> <li>• Disponibilidad de bibliotecas y herramientas _____</li> <li>• Preferencias del cliente _____</li> <li>• Documentación y comunidad _____</li> <li>• Costos asociados _____</li> <li>• Curva de aprendizaje _____</li> <li>• Herramientas y librerías de seguridad _____</li> </ul>

<p>9. ¿Toma en cuenta la sostenibilidad y mantenibilidad del software al momento de elegir lenguajes de programación?</p> <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Sí</li> <li>• <input type="checkbox"/> Medio</li> <li>• <input type="checkbox"/> No</li> </ul>	<p>10. ¿Qué tan determinante considera el propósito específico del proyecto para elegir lenguajes de programación? <b>Favor marcar su respuesta o encerrarla.</b></p> <ol style="list-style-type: none"> <li>1. Nada determinante</li> <li>2. Poco determinante</li> <li>3. Ni determinante ni no determinante</li> <li>4. Determinante</li> <li>5. Muy determinante</li> </ol>
<p>11. ¿Considera necesario tomar en cuenta la experiencia del equipo para elegir lenguajes de programación?</p> <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Sí</li> <li>• <input type="checkbox"/> Medio</li> <li>• <input type="checkbox"/> No</li> </ul>	<p>12. ¿Considera importante la abundancia de librerías y paquetes que permitan facilitar ciertos desarrollos en un lenguaje?</p> <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Sí</li> <li>• <input type="checkbox"/> Medio</li> <li>• <input type="checkbox"/> No</li> </ul>
<p>13. ¿Considera que el rubro de la empresa tiene influencia directa en la elección de lenguajes de programación?</p> <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Sí</li> <li>• <input type="checkbox"/> Medio</li> <li>• <input type="checkbox"/> No</li> </ul>	<p>14. En una escala del 1 al 5, determine la influencia de factores técnicos toma en cuenta para elegir lenguajes de programación.</p> <ul style="list-style-type: none"> <li>• Paradigmas de programación _____</li> <li>• Patrón de diseño de programación _____</li> <li>• Multitasking _____</li> <li>• Nullsafety _____</li> </ul>

**Evaluación de conocimientos de ITIL y metodologías de gestión de proyectos**

15. ¿Ha implementado proyectos donde la elección del lenguaje estuviera influenciada por las prácticas de ITIL?

- Sí
- No
- No lo sé

16. ¿En qué fase del ciclo de vida de ITIL considera más relevante la elección del lenguaje de programación?

- Diseño del servicio
- Transición del servicio
- Operación del servicio
- Mejora continua

17. ¿Qué tan determinante es la alineación de los lenguajes de programación con los estándares y políticas de ITIL en su organización? **Favor marcar su respuesta o encerrarla.**

1. Nada determinante
2. Poco determinante
3. Ni determinante ni no determinante
4. Determinante
5. Muy determinante

18. ¿Qué aspectos del lenguaje considera críticos para la gestión de disponibilidad según ITIL?

- Escalabilidad
- Rendimiento y estabilidad
- Manejo de errores
- Integración con herramientas de monitoreo

19. ¿Qué tan determinante es el soporte del lenguaje para cumplir con las métricas definidas en los acuerdos de nivel de servicio (SLA) en ITIL?

**Favor marcar su respuesta o encerrarla.**

1. Nada determinante
2. Poco determinante
3. Ni determinante ni no determinante
4. Determinante
5. Muy determinante

20. En su empresa, ¿qué tanta influencia tiene las prácticas y principios que propone ITIL? **Favor marcar su respuesta o encerrarla.**

1. Nada influyente
2. Poco influyente
3. Ni influyente ni no influyente
4. Influyente
5. Muy determinante

<p>21. ¿Qué tan relevante es que el lenguaje permita realizar iteraciones rápidas y entregas incrementales como se requiere en metodologías ágiles? <b>Favor marcar su respuesta o encerrarla.</b></p> <ol style="list-style-type: none"> <li>1. Nada relevante</li> <li>2. Poco relevante</li> <li>3. Ni relevante ni no relevante</li> <li>4. Relevante</li> <li>5. Muy Relevante</li> </ol>	<p>22. En una escala del 1 al 5, determine las características que considere útiles de lenguajes de programación que facilite y promueva el trabajo en equipo ágil</p> <ul style="list-style-type: none"> <li>• Facilidad de lectura y mantenimiento del código _____</li> <li>• Disponibilidad herramientas para integración continua (CI/CD) _____</li> <li>• Compatibilidad con pruebas automatizadas _____</li> <li>• Curva de aprendizaje baja _____</li> </ul>
<p>23. ¿En qué medida un lenguaje de programación afecta la velocidad de los ajustes en términos de gestión del cambio en ciclos ágiles? <b>Favor marcar su respuesta o encerrarla.</b></p> <ol style="list-style-type: none"> <li>1. Nada influyente</li> <li>2. Poco influyente</li> <li>3. Ni influyente ni no influyente</li> <li>4. Influyente</li> <li>5. Muy influyente</li> </ol>	<p>24. ¿Qué tan influyente considera un lenguaje de programación con la práctica de integración continua (CI) en un entorno ágil? <b>Favor marcar su respuesta o encerrarla.</b></p> <ol style="list-style-type: none"> <li>1. Nada influyente</li> <li>2. Poco influyente</li> <li>3. Ni influyente ni no influyente</li> <li>4. Influyente</li> <li>5. Muy influyente</li> </ol>
<p><b><u>Evaluación de su experiencia en proyectos de desarrollo de software</u></b></p>	
<p>25. Mencione un proyecto reciente donde la selección del lenguaje fue crítica para el éxito. ¿Qué factores influyeron?, conteste según su experiencia.</p>	<p>26. ¿Ha participado en proyectos que se dificultan debido a la selección del lenguaje de programación?</p> <ul style="list-style-type: none"> <li>• <input type="checkbox"/> Sí</li> <li>• <input type="checkbox"/> No</li> </ul>

<p>27. ¿Considera que la estandarización de un lenguaje de programación facilita la mantenibilidad y escalabilidad de un proyecto o programa de software?</p>	<p>28. En caso de haber tenido la experiencia de integrar múltiples lenguajes de programación en un proyecto, mencione de forma abierta los desafíos y soluciones aplicadas.</p>
<p>29. En proyectos que involucraron varios equipos de trabajo, ¿Cómo garantizo que la elección de lenguajes de programación no obstaculizará el desarrollo del proyecto?</p>	<p>30. ¿Considera que la infraestructura tecnológica (servidores, alojamiento en la nube) influye en la elección de un lenguaje de programación?</p> <ul style="list-style-type: none"><li>• <input type="checkbox"/> Sí</li><li>• <input type="checkbox"/> No</li></ul>

<p>31. ¿Qué tan útil consideraría una rúbrica que permita seleccionar lenguajes de programación? <b>Favor marcar su respuesta o encerrarla.</b></p> <ol style="list-style-type: none"> <li>1. Nada útil</li> <li>2. Poco útil</li> <li>3. Ni útil ni no útil</li> <li>4. Útil</li> <li>5. Muy útil</li> </ol>	<p>32. En escala del 1 al 5, ¿Qué factores considera más determinante para incluir en la rúbrica?</p> <ul style="list-style-type: none"> <li>Soporte y comunidad _____</li> <li>Popularidad _____</li> <li>Rendimiento del lenguaje _____</li> <li>Propósito del proyecto _____</li> <li>Soporte multiplataforma _____</li> <li>Seguridad del lenguaje _____</li> <li>Compatibilidad con otros sistemas _____</li> <li>Rubro de la empresa _____</li> <li>Necesidades del cliente _____</li> <li>Abundancia de librerías y paquetes _____</li> <li>Facilidad de aprendizaje _____</li> </ul>
<p><b><u>Muchas gracias por su apoyo, su respuesta será útil para cumplir con el objetivo de investigación</u></b></p>	

Fuente: Elaboración propia

**Tabla 29 Juicio de Expertos**

A continuación, se presenta una breve encuesta en la participaron varios roles llenando la encuesta. Los valores van desde 0 a 5, siendo 5 el valor de más alto y 0 el más bajo.

Ítem	E1	E2	E3	E4	E5	E6	E7	E8
Sección de Factores	5	4	5	4	3	5	4	5
Sección de Metodologías	2	3	4	2	3	1	3	4
Sección de conocimientos generales	5	4	3	2	5	3	5	1
Sección de ITIL	2	1	4	5	3	2	3	4
Claridad de preguntas	4	3	2	4	3	4	5	2
Facilidad de responder	3	4	5	4	3	4	5	5
Pertinencia de preguntas	3	4	5	4	5	3	2	3
Formato de respuesta correcto	5	4	5	4	5	5	3	3

Fuente: Elaboración propia