

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de
Ingenieros de Sistemas**

**TEMA:
ANÁLISIS CUALITATIVO Y CUANTITATIVO SOBRE ARQUITECTURAS DE
LA INDUSTRIA DE SOFTWARE EN LA PROVINCIA DE PICHINCHA**

**AUTORES:
DENNIS ALFONSO ECHEVERRÍA VILLACÍS
JUAN ANDRÉS REIMUNDO GUALOTUÑA**

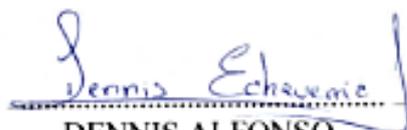
**TUTOR:
GUSTAVO ERNESTO NAVAS RUILOVA**

Quito, febrero del 2020

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Dennis Alfonso Echeverría Villacís, con documento de identificación N° 1723552772 y Juan Andrés Reimundo Gualotuña, con documento de identificación N° 1722128715, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: "ANÁLISIS CUALITATIVO Y CUANTITATIVO SOBRE ARQUITECTURAS DE LA INDUSTRIA DE SOFTWARE EN LA PROVINCIA DE PICHINCHA", mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



DENNIS ALFONSO
ECHEVERRÍA VILLACÍS

CI: 1723552772



JUAN ANDRÉS
REIMUNDO GUALOTUÑA

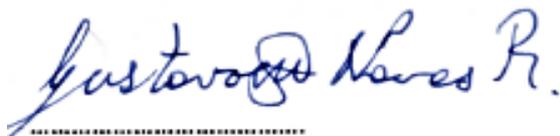
CI: 1722128715

Quito, febrero del 2020

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Artículo Académico, con el tema: "ANÁLISIS CUALITATIVO Y CUANTITATIVO SOBRE ARQUITECTURAS DE LA INDUSTRIA DE SOFTWARE EN LA PROVINCIA DE PICHINCHA" realizado por Dennis Alfonso Echeverría Villacís y Juan Andrés Reimundo Gualotuña, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, febrero del 2020



GUSTAVO ERNESTO NAVAS RUILOVA

CI: 1705675625

Dedicatoria

El presente trabajo investigativo le dedico principalmente a Dios, por darme la vida y la inteligencia para seguir adelante día a día.

A mis padres Fausto y Myrian, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes he logrado llegar hasta aquí y convertirme en un profesional. A mi hermano Rodrigo porque siempre fue mi compañía en los momentos más difíciles y por brindarme su cariño y apoyo incondicional.

Mis tías Glory, Fanny, Mary, ya que, con su cariño, ánimos y afecto siempre me guiaron a pesar de todas las dificultades, les quedaré infinitamente agradecido siempre.

Por último, agradezco a mis amigos que siempre me dieron su amistad incondicional y su ayuda, aún en los peores momentos.

DENNIS

A mis padres María y Juan José que durante toda esta etapa me dieron todo su respaldo, su amor y sus enseñanzas. Gracias por siempre estar pendientes de mi bien estar, por siempre apoyar mis decisiones y por nunca dejarme solo. Gracias a ustedes logre alcanzar esta meta. A mis tíos, tías, primos y primas que siempre me han mostrado su apoyo y afecto. A Ibeth Daniela quien, durante parte de esta etapa de mi vida, me ayudaste, me motivaste y me diste tu amor.

Por último, a mis amigos de la universidad, del cole y de la vida, pues siempre me mostraron toda su apoyo y consideración.

JUAN ANDRÉS

Agradecimiento

Agradecer a Dios por permitirnos formar parte de la UPS, dónde nos formamos como profesionales y a su vez aprendimos a servir a la sociedad.

Agradecer al nuestro tutor Ing. Gustavo Navas, por toda la paciencia mostrada durante este proceso y gracias a su apoyo y sabiduría logramos alcanzar esta meta.

Finalmente, un agradecimiento especial a nuestros docentes, que nos guiaron y apoyaron durante nuestra carrera universitaria.

Análisis Cualitativo y Cuantitativo Sobre Arquitecturas de la Industria de Software en la Provincia de Pichincha

Dennis A. Echeverría¹, Juan A. Reimundo², Gustavo E. Navas³

Resumen-

El presente artículo se enfoca en conocer la situación actual de la arquitectura de software en la provincia de Pichincha, a través de una encuesta que permitirá conocer la opinión de los profesionales en esta área. El objetivo es conocer el nivel actual del uso de la arquitectura de software en Ecuador, en los sectores: gubernamental, empresarial y universitario, que manejan la informática y conocen el estado actual de la arquitectura de software.

Las consultas de investigación se realizaron a través de un cuestionario de preguntas abiertas y cerradas. Estas dieron lugar a un análisis tanto cuantitativo como cualitativo, donde se usó la herramienta Atlas Ti. Las respuestas recibidas muestran la constante evaluación que deriva en el éxito de un proyecto, asimismo la documentación de la arquitectura se efectúa con el fin de registrar los elementos que conforman el diseño de la solución para futuras mejoras o cambios en la industria del software con metodologías ágiles

Finalmente, se recomienda buenas prácticas en el diseño de la arquitectura de aplicaciones ya que esta sostendrá todas las funciones y permitirá actualizar de forma rápida y efectiva futuras mejoras o modificaciones.

Palabras Claves: Metodologías Ágiles, Informática, Documentación, Diseño y Atlas Ti ©.

Abstract-

This article focuses on knowing the current situation of software architecture in the province of Pichincha through a survey that will establish the point of view of professionals in the area. The objective is to know the current level of the use of software architecture in Ecuador, in all areas of the Government sector, Businessmen and Universities that handle in the area of computer science and from the point of view of the impact and development of the software architecture, as well as the current state of the software architecture.

The research consultations were conducted through a questionnaire of open and closed questions. These resulted in both quantitative and qualitative analysis using Atlas Ti. The responses received show the constant evaluation that results in the success of a project, also the documentation of the architecture is carried out in order to record the elements that make up the design of the solution for future improvements or changes in the software industry whit agile methodologies.

With this in mind, good practices in the design of the application architecture are recommended as this will support all functions and will allow future improvements or modifications to be updated quickly and effectively.

Keywords: Agile Methodologies, Software Industry, Documentation, Design and Atlas Ti ©.

¹ Estudiante de Ingeniería de Sistemas – Universidad Politécnica Salesiana – Sede Quito. decheverria@est.ups.edu.ec

² Estudiante de Ingeniería de Sistemas – Universidad Politécnica Salesiana – Sede Quito. jreimundo@est.ups.edu.ec

³ Máster en Software Libre en la Universidad Abierta de Cataluña. gnavas@ups.edu.ec

1. Introducción

Un aspecto importante en el crecimiento de un país es, el avance de su tecnología que permite su desarrollo económico y social. En este contexto, la tecnología favorece considerablemente, que la industria alcance una mayor productividad y permite mejorar el nivel de calidad de vida de las personas. En este sentido, las empresas le han dado gran importancia al impacto positivo que produce la inversión en tecnología, de tal manera, que en la actualidad se dispone de personal profesional cualificado, dedicado al desarrollo de la arquitectura que se emplea en la industria de software, en la provincia de Pichincha de la República del Ecuador.

Es importante resaltar que, la arquitectura de software juega un papel primordial en cualquier proyecto de desarrollo, pues representa de forma definitiva el corazón del programa debido a que está enmarcada desde los algoritmos hasta la estructura de los datos. Viene a ser un esquema de trabajo que define el tipo de código fuente, las pautas, objetivos del desarrollo y definirá la forma de interacción de los diferentes elementos del sistema.

Otros elementos importantes dentro de la arquitectura de software y que el profesional de informática debe conocer en profundidad, es la forma como el sistema de información manejará lo relativo a la seguridad, así como la comunicación entre componentes, el acceso a fuente de datos y el formato de los datos, entre otros.

1.1 Trabajos Relacionados

Para Colombia la industria de software es considerada como sector de talla mundial en el desarrollo de software empaquetado, contribuyendo al desarrollo económico y productivo de ese país, considerado ejemplo para el resto de Latinoamérica [1], esta investigación hizo un análisis en la producción y desarrollo de software en el

país Neogranadino que contribuyo a esta investigación de forma positiva ya que encauso los modelos de preguntas que se establecieron dentro de la encuesta a realizar.

En este orden de ideas, la tendencia ascendente en el desarrollo de la industria de software, es un aspecto fundamental que el estado ecuatoriano ha tomado en cuenta para la transformación y crecimiento de la matriz productiva. Según datos de estudios realizados por diversos organismos públicos y privados, señalan que la producción de programas informáticos, ha crecido significativamente en los últimos años. Ante esto, las empresas del ramo se plantean algunas interrogantes que dan pie a la búsqueda de acciones y actuaciones en este particular, tales como: ¿Se han generado cifras formales y públicas que indiquen el crecimiento del sector software y la exportación de este producto hecho en Ecuador? ¿Existen registros de datos formales que sirvan como soporte del apoyo por parte del Estado a la industria de este sector? En este contexto hay que destacar que, esta investigación abordó el tema de la evolución y desarrollo de software en el Ecuador, dicho trabajo fue un marco de referencia en esta investigación ya que este estudio fue muy similar y permitió diseñar la estrategia metodológica que se utilizó en el presente estudio [2].

De igual forma para el presente trabajo de investigación se consideró la opinión de profesionales en el área de la informática, quienes fueron abordados con la finalidad de conocer a profundidad *¿Cuál es el estado o practica actual de las arquitecturas de software en la provincia de Pichincha?* y gracias al apoyo de la empresa pública y privada, así como de universidades, se pudo lograr el desarrollo de la investigación, la cual permitió recopilar importante información que dio como resultado la identificación del estado actual de la arquitectura de software en el Ecuador, así

como requisitos y metodologías aplicadas al desarrollo de aplicaciones.

2. Metodología

En esta sección, se describen los aspectos metodológicos utilizados en la presente investigación, empezando por el diseño, la población y la muestra. Luego se describen los instrumentos usados, así como también las categorías y descripción de los métodos aplicados y finalmente, los procesos realizados para la recolección de la información.

Es importante resaltar que el desarrollo de esta investigación, tiene una combinación entre metodologías orientadas en dos enfoques: el enfoque cualitativo que permitió recopilar información verbal en las entrevistas y que se registraron en forma de audio para, posteriormente, transcribirlas a textos, con la finalidad de entender a profundidad los hechos o acontecimientos y luego analizarlos de forma interpretativa. Por otro lado, en enfoque cuantitativo, con el tratamiento de la información numérica, que permite explicar en cifras lo que se observa [3]. De igual manera, se indica que una investigación mixta permite encontrar respuestas que no se encuentran cuando se plantea una única metodología [4]. De la misma forma, señala que una ventaja de emplear diseños mixtos es recabar información más profunda por lo que los resultados adquieren mayor validez lo que determina que estas metodologías son complementarias [5].

2.1 Diseño

En el diseño de la investigación se describen, los procedimientos empleados para recoger la información, analizarla y a su vez interpretar los datos obtenidos, recordando que desarrollo de esta, se fundamentó de una manera tanto cualitativa como cuantitativa. En tal sentido, los datos fueron recogidos por medio de entrevistas, que fueron transcritas

de audio a texto, con preguntas abiertas y cerradas y para el análisis cualitativo de los datos obtenidos se realizó con la herramienta Atlas TI mientras que lo cuantitativo fue mediante la estadística descriptiva.

2.2 Población y Muestra

La población objeto de estudio, fue el universo de personas con experiencia en el campo de desarrollo de software, al contar con una población muy pequeña, la entrevista se realizó a un grupo de personas con experiencia en el campo del desarrollo de software en empresas tanto públicas o privadas sin límites de edad, género, nacionalidad o años de experiencia.

El estudio, permitió obtener información más precisa sobre el uso y la aplicación de las arquitecturas de la industria de software. en la provincia de Pichincha.

2.3 Técnicas de Recolección de Datos

Para recolectar los datos en la investigación se aplicó la técnica de grupos de enfoque y entrevistas semi-estructuradas. La primera técnica, permitió que la investigación se centrara en una cantidad de participantes reducida, pero que cumplieran con las características orientadas a los objetivos de la investigación. Estos grupos de enfoque, permitieron analizar aspectos relevantes del estudio, especialmente en la aplicación de las arquitecturas de la industria de software en la provincia de Pichincha.

Por otro lado, las entrevistas realizadas a este grupo de enfoque, permitieron recabar la información que contribuyó a dar cumplimiento a los objetivos de la investigación, facilitando complementar la información con experiencias personales. En tal sentido, las preguntas de las entrevistas fueron diseñadas previamente por los investigadores, sin embargo, debido a la dinámica y a las necesidades del estudio, se formularon preguntas que no se encontraban planteadas pero que se consideraron necesarias, con el fin de

obtener mayor información sobre algún tema tratado en el momento de la entrevista.

2.4 Fases de la investigación

La investigación inició con una revisión bibliográfica, empleando fuentes primarias y secundarias. En esta etapa, se desarrollaron actividades como: revisión de trabajos realizados anteriormente por distintos autores, análisis de los trabajos revisados y el planteamiento del problema de investigación.

De igual forma, para el inicio del trabajo en campo se realizó una exploración de empresas públicas y privadas dedicadas al desarrollo de software mediante la herramienta de Google Maps. Seguidamente, se realizó un acercamiento a estas organizaciones solicitando apertura para realizar la investigación, efectuando un primer contacto con el personal a participar en el proyecto de investigación, al aplicar los instrumentos, se procedió con la transcripción y tabulación de la información recabada y la socialización de resultados preliminares.

2.4.1 Acercamiento inicial

Una vez ubicadas las firmas que se dedican al desarrollo de software mediante la herramienta Google Maps, se seleccionaron compañías que se encuentren en un perímetro accesible para los investigadores. Estas, facilitaron desde su infraestructura, personal y además, la información necesaria para el desarrollo de la investigación, además, se realizaron solicitudes de permisos y accesos a las instalaciones. En algunos casos las solicitudes fueron de forma escrita y en otras de forma oral, siendo estas últimas en compañías que dieron una apertura inmediata para la investigación.

3. Análisis de Resultados

3.1 Análisis de resultados cuantitativos

A continuación, se presenta los resultados obtenidos del análisis cuantitativo de los resultados de la información proveniente de veinte profesionales considerados clave, para la investigación. Se aplicó un cuestionario estructurado, integrado por nueve preguntas cerradas. Esta actividad se realizó, con la finalidad de recopilar información adicional que contribuya a reforzar el análisis cualitativo, las opiniones recopiladas fueron referente al tema del software, arquitectura utilizada, tecnologías, lenguajes de programación y metodologías. Respuestas del cuestionario aplicado:

Tabla 1. Arquitecturas más usadas

Alternativas	Frecuencia	Porcentaje
Arquitectura en capas	19	42%
Arquitectura de repositorio	3	7%
Arquitectura cliente-servidor	19	42%
Arquitectura de tubería y filtro	4	9%
Totales	45	100%

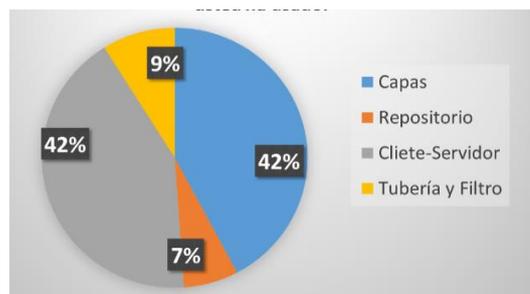


Figura 1. Distribución Porcentual de la Muestra según Arquitecturas más usadas

Análisis

El 42% de los encuestados, manifiesta que, utilizan la arquitectura cliente servidor, el 42% de los encuestados, indica utilizar la arquitectura en capas, el 9% de los encuestados, comentó que usan la arquitectura de tubería y filtro, y finalmente

el 7% restante que refiere trabaja con la arquitectura de repositorio.

Interpretación

En función a los resultados obtenidos, se puede inferir en base a la percepción de los especialistas, que la arquitectura cliente servidos y la arquitectura en capas, poseen gran versatilidad y sus características de uso, favorecen el desenvolvimiento y la creatividad del diseñador, además es posible que por políticas de la empresa donde estos estén laborando y del entorno que se esté desarrollando la actividad, influyan de forma significativa en su elección para su uso.

Tabla 2. Definición de una arquitectura para todas las aplicaciones en el desarrollo de software

Alternativas	Frecuencia	Porcentaje
SI	5	25%
NO	15	75%
Totales	20	100%

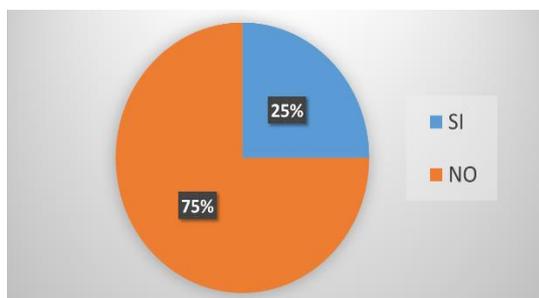


Figura 2. Distribución Porcentual de la Muestra según la Definición de una arquitectura para todas las aplicaciones en el desarrollo de un software.

Análisis

El 75% de los encuestados manifiesta que, no tiene definida una arquitectura para cada tipo de aplicación, mientras el 25% manifestó lo contrario.

Interpretación

Los resultados arrojados pueden ser interpretado como que la mayoría de los diseñadores no consideran importante

aplicar una arquitectura específica a las aplicaciones o que las aplicaciones que emplean, pueden ser usadas con facilidad sin tener que asignar un tipo de arquitectura, en todo caso, está en función a las modalidades de trabajo, a la preparación y conocimientos del profesional en el área.

Tabla 3. Uso de tecnología en versiones actuales en el desarrollo de aplicaciones

Alternativas	Frecuencia	Porcentaje
SI	13	65%
NO	7	35%
Totales	20	100%

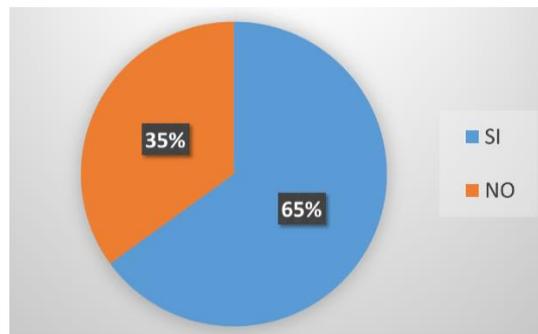


Figura 3. Distribución Porcentual de la Muestra según el Uso de Tecnologías Actuales para el desarrollo de aplicaciones.

Análisis

El 65% de los encuestados manifiesta que, si utiliza tecnología en versiones actuales, mientras el 35% manifestó lo contrario.

Interpretación

De nuevo, las opiniones de la mayoría permiten inferir que, los profesionales especialistas en el área están consiente que, al usar nuevas tecnologías, obtendrán mayor provecho en sus resultados y por ende, presentaran un aporte considerable para la organización donde laboran, con la finalidad de mejorar tanto la calidad como la productividad de la misma.

A. Sistemas operativos

Tabla 4. Principales tecnologías que usa para el desarrollo de las aplicaciones

Alternativas	Frecuencia	Porcentaje
Linux	18	41%
Windows	16	36%
Mac	10	23%
Totales	44	100%

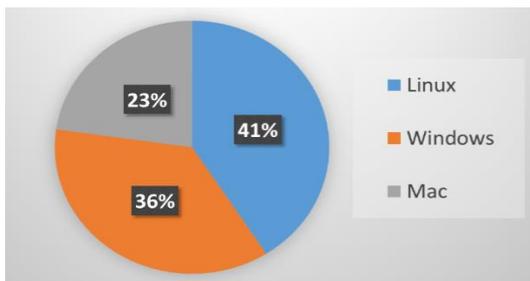


Figura 4. Distribución Porcentual de la Muestra según las Principales tecnologías usadas para el desarrollo de aplicaciones.

Análisis

Como se puede identificar, el 41% manifiesta que, utiliza el sistema operativo Linux para el desarrollo de aplicaciones, mientras que el 36% de los encuestados manifestó, que utiliza Windows para el desarrollo de sus programas, adicionalmente un 23% indicó que utiliza el sistema operativo Mac.

Interpretación

En este apartado se visualiza que, según la mayoría de los encuestados, el sistema Linux representa una alternativa de mayor versatilidad en cuanto a sus bondades, no obstante, el sistema operativo Windows se percibe como una opción aceptable y entre ambos sistemas operativos, dominan la preferencia de los especialistas en esta área. Por otro lado, también puede estar asociado a que las empresas para la cual laboran, cuentan solo con uno de los dos o los dos

sistemas y esto promueva su uso con mayor aceptación.

B. Lenguajes de programación

Tabla 5. Principales tecnologías que usa en el desarrollo de las aplicaciones para lenguajes de programación

Alternativas	Frecuencia	Porcentaje
.Net	06	10%
JAVA	16	25%
JavaScript	15	24%
PHP	11	17%
Ruby	06	10%
Python	07	11%
Perl	02	03%
Totales	63	100%

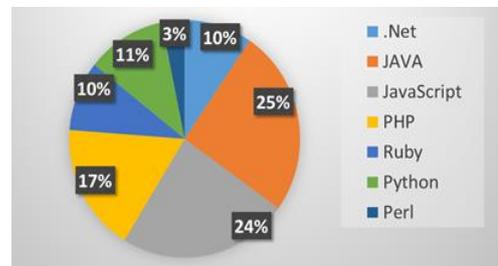


Figura 5. Distribución Porcentual de la Muestra según las principales tecnologías que usa en el desarrollo de lenguajes de programación.

Análisis

Se puede identificar que el 10% manifiesta que, utiliza .NET como lenguaje de programación, por otra parte, el 25% de los encuestados indica que utiliza JAVA como lenguajes de programación, seguidamente, un 24% de los encuestados, dijo que utiliza JavaScript como lenguaje de desarrollo, en tanto, el 17% de los que participaron en la encuesta manifestaron que utilizan PHP para la construcción de aplicaciones, al igual que el 10% de los encuestados utiliza Ruby, en tanto el 11% de los profesionales consultados utilizan Python como lenguaje de programación y finalmente el 3% de los encuestados indico utilizar Perl para diseñar sus entregables.

Interpretación

Los resultados permiten al investigador observar que JAVA y JavaScript son los

lenguajes de programación de mayor uso en porcentajes individuales y ambos acumulan casi el 50% de los resultados, Esto parece indicar que poseen mayores y mejores condiciones de rendimiento y efectividad al momento de su uso y por ello, gozan de mayor preferencia por parte de los especialistas en el área. Por otro lado, hay que considerar también, que sus respuestas pudieran estar asociadas al tipo de proyecto que estos estén desarrollando en un determinado momento por las exigencias del cliente o por las condiciones de disponibilidad con que cuente la empresa que los contrate, en todo caso, aparentemente, están por encima de los restantes lenguajes.

C. Base de datos

Tabla 6. Principales tecnologías usadas en el desarrollo de las aplicaciones para Base de datos

Alternativas	Frecuencia	Porcentaje
PostgreSQL	16	31%
MySQL	18	35%
MS-SQL	09	18%
No-SQL	08	16%
Totales	51	100%

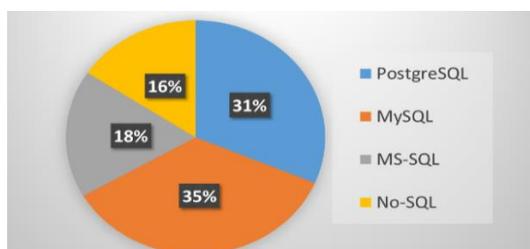


Figura 6. Distribución Porcentual de la Muestra según las principales tecnologías usadas en el desarrollo de las aplicaciones para bases de datos.

Análisis

El 31% de los encuestados manifiesta que utilizan PostgreSQL para el desarrollo de Base de datos, por otra parte, un 35% de los encuestados, dijo que usa MySQL para

trabajar con Base de datos, mientras que, el 18% de los que participaron en la encuesta manifestaron que trabajan con MS-SQL para manipular las Bases de datos, por su parte, el 16% de los encuestados maneja No-SQL para la administración de Base de datos.

Interpretación

Los resultados parecen indicar que MySQL y PostgreSQL son las tecnologías que poseen la mayor atención y atracción por parte de los especialistas, cada una con más del 30% de las preferencias lo que parece indicar que para el desarrollo de bases de datos, estos dos poseen condiciones superiores en cuanto a versatilidad, rendimiento, disponibilidad y facilidades en relación al resto de los que se consideraron en este estudio, lo que permite inferir que tanto los clientes como las organizaciones, promuevan y prefieran su uso, hecho que los coloca en ventaja sobre MS-SQL y el No-SQL.

Tabla 7. Separación Obligatoria de las funcionalidades del Software

Alternativas	Frecuencia	Porcentaje
SI	18	90%
NO	2	10%
Totales	20	100%

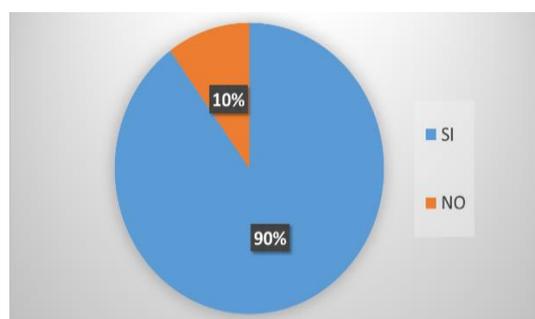


Figura 7. Distribución porcentual de la muestra según la separación obligatoria de las funcionalidades del Software.

Análisis

El 90% de los encuestados, manifiesta que, el software debe separar sus funcionalidades, mientras el 10% indica lo contrario.

Interpretación

En este punto las opiniones de los especialistas parecen indicar que es de suma importancia para desarrollar un trabajo seguro y productivo, usar un software que permita separar sus funcionalidades, lo que los puede llevar a la selección de aquellos softwares que les brinde esta condición.

Tabla 8. Calificación sobre la acción de remplazo de las versiones anteriores de software

Alternativas	Frecuencia	Porcentaje
Muy Bueno	11	55%
Bueno	07	35%
Aceptable	02	10%
Malo	0	0%
Muy Malo	0	0%
Totales	33	100%

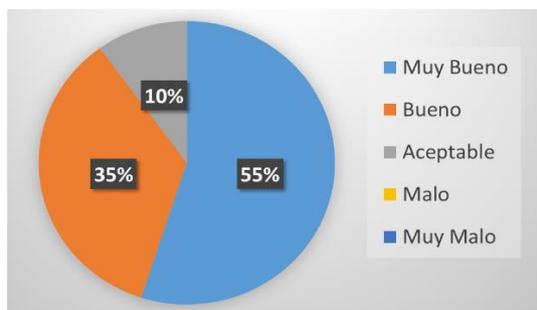


Figura 8. Distribución porcentual de la muestra según la calificación sobre la acción de remplazo de las versiones anteriores de software

Análisis

El 55% de los encuestados, manifiesta que, es muy bueno el remplazo de las versiones anteriores de software, mientras el 35% indica que es bueno el remplazo de las versiones anteriores de software, seguidamente él, 10% comenta que es

aceptable el remplazo de las versiones anteriores de software, mientras que, malo y muy malo no recibieron opiniones.

Interpretación

En función a los resultados se puede inferir que los especialistas consideran de suma importancia el remplazo de versiones anteriores debido a que, para dar soluciones efectivas y confiables ante un proyecto de envergadura, es relevante contar con la última tecnología que se adapte a las necesidades y requerimientos del mismo y así garantizar un resultado de calidad.

Tabla 9. Ventajas de diseñar y documentar la arquitectura de software

Alternativas	Frecuencia	Porcentaje
Permite que la ejecución del software sea más sencilla.	3	15%
Es muy útil para futuras modificaciones.	3	15%
Es útil para el estudio de mejoras continuas.	14	70%
Totales	33	100%



Figura 9. Distribución porcentual de la muestra según las ventajas de diseñar y documentar la arquitectura de software

Análisis

El 15% de los encuestados, manifiesta que, permite la ejecución del software sea más sencilla, al igual que otro 15% indica que es muy útil para futuras modificaciones y finalmente la mayoría representada en un

70% opina que es útil para el estudio de mejoras continuas.

Interpretación.

La mayoría de los encuestados consideró que la documentación es útil para mejoras continuas, esto parece indicar que está bien comprendido que mientras se tenga la en documentos la información ordenada, se tendrá mayor confianza y se tendrá una menor probabilidad de obtener fallos no deseados de imprevistos.

Tabla 10. Metodologías Ágiles de Mayor uso

Alternativas	Frecuencia	Porcentaje
Scrum	20	54%
Extreme Programming (XP)	8	22%
KanBan	9	24%
Totales	37	100%

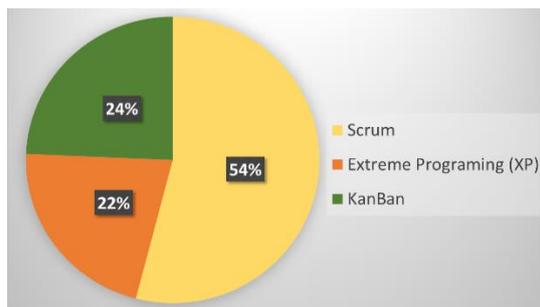


Figura 9. Distribución porcentual de la muestra según la metodologías ágiles de Mayor Uso.

Análisis

El 54% de los encuestados, manifiesta que, la metodología ágil que utiliza es Scrum, seguida el 22% de los encuestados indica que, la metodología agile que utiliza es Extreme Programming (XP), seguida del 24% que comentan utilizar KanBan.

Interpretación.

Como se puede observar, según los especialistas encuestados, la metodología Scrum gaza de la mayor preferencia en su

selección entendiéndose que les ofrece condiciones positivas por encima de las demás, pero también puede estar asociado a las exigencias del proyecto o de las disponibilidades tecnológicas que posea la organización.

Tabla 11. Bases de las Metodologías ágiles

Alternativas	Frecuencia	Porcentaje
Estar orientada al negocio	3	15%
Ser flexible	16	80%
Ser colaborativa	1	05%
Totales	20	100%

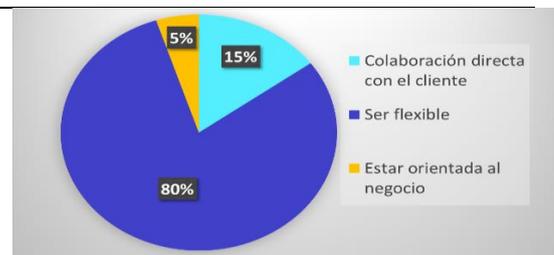


Figura 10. Distribución porcentual de la muestra según las Bases de las metodologías ágiles.

Análisis

El 5% de los encuestados, manifiesta que, las bases de la metodología ágil están orientadas al negocio, seguida de un 80% que indica, las bases de la arquitectura ágil deben ser flexibles, seguidamente, el 15% de los encuestados indica que, las bases de la metodología se encuentran en ser colaborativa.

Interpretación.

En esta oportunidad, se puede inferir que las bases de las metodologías ágiles que más llama su atención son las que poseen características flexibles, esto puede estar asociado a las facilidades y comodidades que este tipo de metodologías brinda al momento de desarrollar un determinado proyecto.

3.2 Análisis Cualitativo

En el siguiente apartado, se presenta el análisis inductivo de los resultados de la

información, proveniente de veinte informantes considerados clave, para la investigación. Se aplicó una entrevista semi estructurada, integrada por veintiún preguntas abiertas, las mismas que se transcribieron de audio a texto. Esta actividad, se realizó con la finalidad que cada informante pudiese expresar y explicar sus opiniones, referente al tema de software, usos, aplicaciones y el manejo tecnológico y todo lo que ello implica. En cada pregunta, se manejaron categorías que orientaron la investigación, referente a la arquitectura de software en el campo de la informática.

Cabe destacar, que para procesar la información se utilizó el programa ATLAS.Ti, donde emergieron un grupo de cuarenta y ocho códigos, los cuales, según su frecuencia se agruparon en seis categorías que son la arquitectura de software, metodologías ágiles, funcionamiento de la arquitectura de software, evolución de software, evaluación de la arquitectura de software y estado o práctica actual de

software, en la Provincia de Pichincha en Ecuador.

De este proceso, se desprenden hallazgos, producto de las vivencias y experiencias de los informantes claves, los cuales fueron referencia para determinar el diseño, uso y aplicación de software como un producto intangible, en el cual intervienen especialistas de la ingeniería informática y de la arquitectura de software.

En primer lugar, emerge la red semántica “Arquitectura de software”, integrada por los siguientes códigos: calidad y diseño del software, metodología para elegir la arquitectura y capacidad, componentes, costos y características del software; al igual que la elección errónea de la arquitectura.

Al describir la primera red, que lleva por título “Arquitectura de Software”, ésta se define como el diseño técnico y funcional o nivel estructural del sistema. El diseño de este producto. va a “depender de la necesidad y cantidad de usuarios. Entiéndase, por arquitectura de software, “a un grupo de abstracciones y patrones que

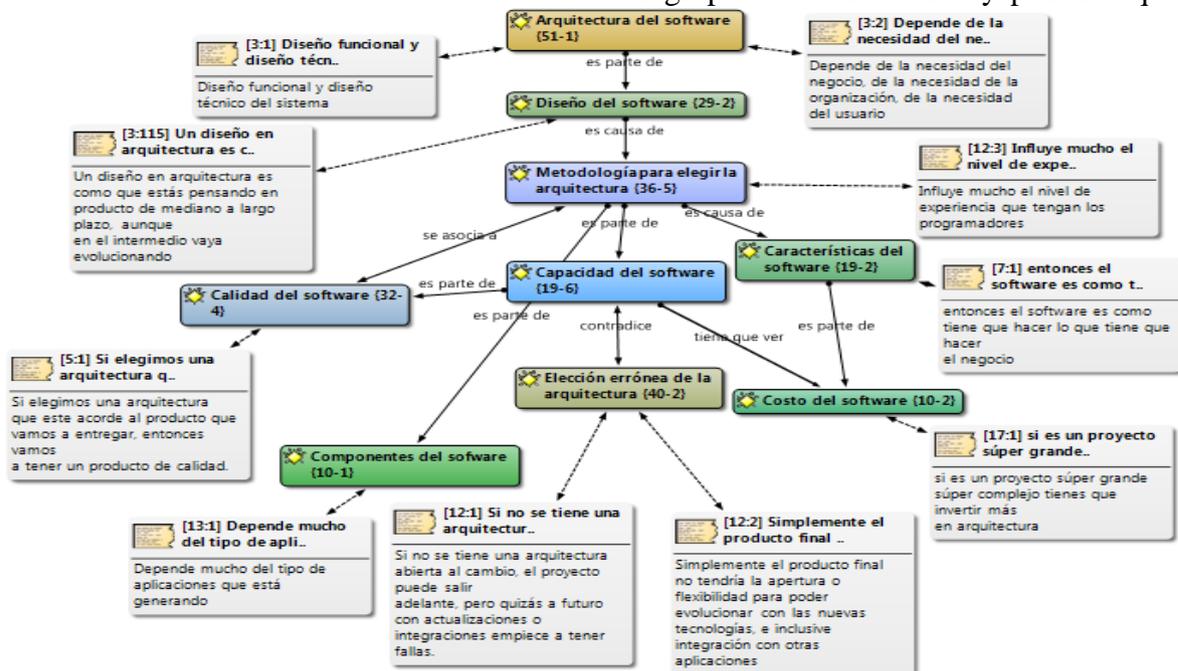


Figura 11. Primera red semántica: Arquitectura de software.

brindan un esquema de referencia útil, para el desarrollo de software dentro de un sistema informático.” [6]. Para el (Informante 2, cita 28), sobre la arquitectura de software, señala que, “*es el diseño funcional y diseño técnico del sistema*”.

Referente al diseño del software, los componentes o unidades modulares que se apliquen, juegan un papel determinante para el conjunto de servicios y funciones, a la que va estar destinado. En el diseño, se debe tener en cuenta la usabilidad del software, al igual que la seguridad para lo cual, es necesario definir a qué tipo de usuarios va dirigido; aunque estos en su totalidad no se conozcan. Referente al diseño, éste debe proyectarse, “*Un diseño en arquitectura, es cómo estás pensando el producto a mediano o largo plazo*”. (Informante 2, cita 83). El diseño es lo que va a integrar el proyecto. Es el plano, para que el equipo técnico tenga claridad. De allí, la importancia de su adecuación y la modularidad; es el producto que va a portar la solución a la demanda del cliente.

En cuanto a los Métodos para elegir la arquitectura de software, la experiencia es el método común de referencia. Es influyente el nivel de experiencia de los programadores; de acuerdo a este, se elige la arquitectura que más se adapte a las necesidades del cliente, con documentación para el soporte, según su aplicación.

En lo que refiere a la calidad, un aspecto que destaca tiene que ver con la capacidad de evolucionar, depende en gran manera de los resultados que obtenga el cliente. “*Sí hicimos una arquitectura que esté acorde al producto que vamos a entregar, entonces vamos a tener un producto de calidad.*” (Informante 4, cita 17). “*La calidad del software, evalúa la arquitectura, si ha cumplido con todos los patrones o todas las características que necesita tener.*” (Informante 8, cita 7). “*Si una arquitectura está bien seleccionada, va a terminar en un*

producto de calidad y sobre todo probado” (Informante 9, cita 18).

Los conocimientos de los componentes del software, dependen de la magnitud del proyecto, ya que este puede evolucionar y cambiar. “*Los componentes dependen del tipo de aplicaciones que se están generando.*” (Informante 20, cita 27). Referente a la capacidad del software, su propósito es “*optimizar los atributos comunes de calidad como rendimiento, seguridad y capacidad de administración.*” (Informante 1, cita 17). Debe existir una evaluación constante, de la calidad del producto, allí se puede hacer comparaciones con arquitecturas de referencia. Al referirse a la calidad y capacidad del software, para el (Informante 2, cita 10), “*Es que el sistema, tenga capacidad de ser escalable y poder evolucionar*”.

En cuanto a las características y costos, estas se relacionan con la capacidad del software y depende de la magnitud. Si es un proyecto grande, la inversión debe ser mayor. Ello depende, de los requerimientos funcionales que necesita el cliente. “*En la actualidad, las empresas grandes emplean arquitectura basada en rendimiento y seguridad, como la arquitectura de capas, lo cual implica un mayor costo de implementación*” (Informante 1, cita 11). “*Las características del software, responden a las solicitudes o necesidades que tiene una empresa o sus puntos de negocio.*” (Informante 14, cita 1). Estas dependen de las características de la población a quien se ofrece el servicio.

Por su parte, la elección errónea del software, refiere a diversos aspectos entre los que están la funcionalidad. Si esta es deficiente, puede conducir al fracaso del proyecto. De allí, la importancia de seleccionar una buena base de la arquitectura, para que el software desarrolle un funcionamiento óptimo. Estas decisiones no adecuadas, pueden hacer que se tenga que rehacer de nuevo el proyecto,

incrementando gastos, y demanda de más tiempo. Una base de arquitectura funcional, evita gastos innecesarios. “*Tú software no va a funcionar; es probable que fracase a futuro.*” (Informante 5, cita 17). “*El producto final no tendría apertura, o flexibilidad para poder evolucionar con las nuevas tecnologías, e inclusive al integrar otras aplicaciones.*” (Informante 10, cita 22).

La segunda red semántica que se construye, tiene como título las “Metodologías Ágiles”. En la cual, emergieron las siguientes unidades de análisis: metodologías empleadas, elección del software, scrum, efectividad de las metodologías ágiles y participación del cliente. Estos códigos, se asocian a las metodologías que se definen como ágiles, buscan dar respuesta a las necesidades del diseño y operatividad del software.

La segunda red semántica, se titula metodologías ágiles. Define estas acciones como las prácticas empleadas en la arquitectura de software, que reúnen los requerimientos técnicos y operacionales. Se

compone de los elementos del software y las relaciones entre ellos. Las metodologías ágiles, propician el desarrollo de proyectos flexibles, que se adaptan a los cambios y necesidades de los clientes, en correspondencia a los objetivos de las empresas. Las metodologías ágiles, están centradas en conformar un equipo colaborativo de trabajo, integrado por los diseñadores del software y la participación del cliente, al aportar experiencias y plantear las necesidades; para lograr la adaptabilidad y poder ofrecer un producto de máxima calidad [7].

Las metodologías ágiles, son vistas como un método de trabajo en equipo. Los expertos en diseño del software, se enfocan en satisfacer las necesidades de los clientes, al mantenerles informados de los avances y logros de los proyectos.

Este método, por lo general se desarrolla en etapas, en las cuales el equipo diseñador aporta sus conocimientos tecnológicos y motivación; aunado a la experiencia del cliente, lo que constituye, un importante potencial para concretar los objetivos.

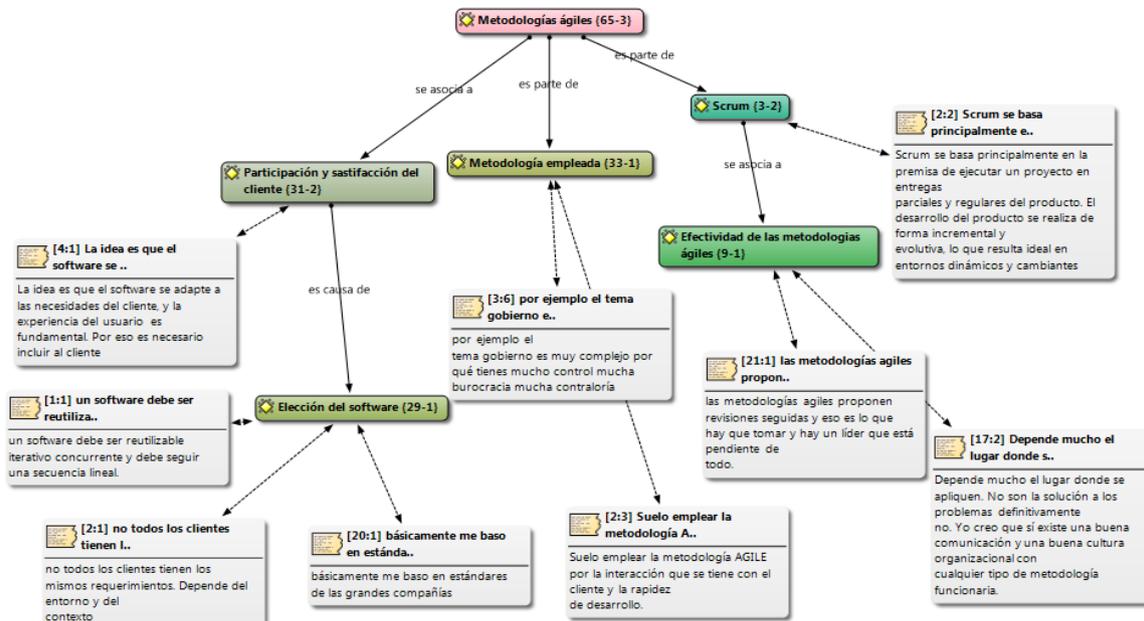


Figura 12. Segunda red semántica: metodologías ágiles.

Referente a la arquitectura de software con metodologías ágiles, el (Informante 3, cita 27), señala que, *“Las prácticas ágiles te ayudan a realizar una buena arquitectura de software, porque tienes un diseño para realizar el proyecto, y lo planificas de una manera. Ese tema de los Sprint, constantemente pasa revisión y te dice si la arquitectura es correcta o no”*. Entre las metodologías más utilizadas por los informantes, están las ágiles, que son de fácil aplicación.

Estas metodologías, propician el ahorro de tiempo y dinero, al trabajar sobre plazos de entrega consensuado con el cliente, a quien se le hace una entrega paulatina y funcional del producto; donde la satisfacción del cliente se convierte en un indicador de evaluación del producto. Estos resultados, permiten hacer cambios o adaptaciones; lo que ayuda a mejorar el producto y acercarse lo más posible a las necesidades empresariales. Todo ello, basado en la interacción con el cliente; ya que, de acuerdo al proyecto a desarrollar, es importante adaptarse a los estándares que solicita el usuario. Es importante resaltar, que las metodologías ágiles, son parte de un entorno tecnológico cambiante y dinámico, que más se utiliza en el Ecuador.

Por su parte Scrum, es una metodología que está constituida por pequeños proyectos, que se desarrollan por fases; los que conforman el producto final. Este método parte del análisis, para desarrollar el proyecto y luego evaluarlo. Scrum, es una metodología ágil, permite la flexibilidad, productividad y rapidez, al aplicar el respectivo software. Para el (Informante 1, cita 46), *“Scrum, se basa principalmente en la premisa de ejecutar un proyecto en entregas parciales y regulares del producto (...) resulta ideal en entornos dinámicos y cambiantes”*.

Por su parte, la revisión del código, *“te ayuda a socializar el código que estas generando, cuando tu revisas códigos tu*

estas entendiendo que es lo que hace otra persona, y también es como si tú puedes aportar ideas” (Informante 20, cita 68). De igual forma, es importante que se cumpla con la codificación, en cuanto al estilo, uso y funcionalidad; todo ello, para verificar el cumplimiento de la arquitectura de software. Para el (Informante 9, cita 38) *“es muy importante revisar el código y hacerle pruebas de funcionamiento.”*

Es de gran relevancia, realizar las revisiones del código: pues este, permite corregir errores y evolucionar continuamente. Por su parte, la elección del software, *“depende del problema que se quiere resolver.”* (Informante 19, cita 57). *“Básicamente, me baso en estándares de las grandes compañías.”* (Informante 18, cita 6), y para el (Informante 11, cita 1), *“Un software debe ser reutilizable, interactivo, concurrente y debe seguir una secuencia lineal”*. Es importante, seleccionar y hacer un diseño, que permita la evolución.

La participación del cliente, es parte de las metodologías ágiles. El cliente, es una fuente importante de información, basada en las experiencias sobre las necesidades e intereses, de la empresa o negocio en particular.

De allí, que es considerado clave para el diseño, desarrollo y éxito del proyecto. Para el (Informante 20, cita 84), *“Tener al cliente junto a ti, las ideas que ellos te dan pueden ir cambiando, para mejorar el software”*. Asimismo, *“La comunicación con los clientes, el trabajar juntos por un objetivo común.”* (Informante 19, cita 119). El principal propósito de un equipo técnico en arquitectura de software, es lograr la satisfacción del cliente; mediante la entrega constante a tiempo, con calidad y funcionalidad de los proyectos que componen el software.

Para el (Informante 1, cita 48), *“las prácticas ágiles, emplean arquitecturas de software que reúnen todos los requerimientos técnicos y operacionales”*,

al respecto [8] plantea que “Las metodologías ágiles, son una excelente alternativa para guiar proyectos de desarrollo de software”.

La tercera red semántica, refiere al “Funcionamiento de la arquitectura de software”; en ésta emergen los códigos: calidad y capacidad del software, cohesión, acoplamiento, documentación de la arquitectura y estándares de calidad. Ver red semántica 3.

El funcionamiento corresponde a la automatización de la tarea. “La funcionalidad va afectar el tema de organización, el código va afectar la interacción de los módulos de un sistema; definitivamente, va afectar la calidad del producto.” (Informante 20, cita 40). De allí, la importancia que el diseño y desarrollo se evalúe constantemente, para obtener logros efectivos. La funcionabilidad, busca evitar los efectos negativos en cadena.

Por su parte, la calidad del software según los modelos y estándares, como la norma ISO/IEC 9126, debe caracterizarse por la calidad del producto, eficiencia, confiabilidad, facilidad de mantenimiento, funcionalidad, usabilidad y la portabilidad. Para [9], “Los estándares más completos

son la ISO/ IEC 9126 y 25010, pues son estándares mixtos, con un propósito general-reutilizable, con aplicabilidad en casi todo tipo de proyecto, capaces de evaluar la calidad interna, externa y en uso”. Para el (Informante 2, cita 10), al referirse a la calidad del software plantea que, “es que este sistema tenga capacidad de ser escalable, y de poder evolucionar”. Para el (Informante 8, cita 6), “la calidad de arquitectura de software, tú la evalúas, si los componentes cumplen con las especificaciones del Software.” La capacidad se evalúa, con el número de funcionalidades que se van agregando. Sí el número de funcionalidades es reducido, la capacidad puede ser baja. Para obtener productos de calidad, debe evaluarse la autorregulación, porque su estructura, tiende a ser más compleja en la medida que se desarrolla o crece el programa.

El documentar la arquitectura, consiste en la diagramación y registro de todos los elementos y estructuras, que integran el sistema que componen el software. De igual forma, se registra que tipo de relación e interacción existe, entre los componentes de la arquitectura del computador, para satisfacer tanto los objetivos de la empresa

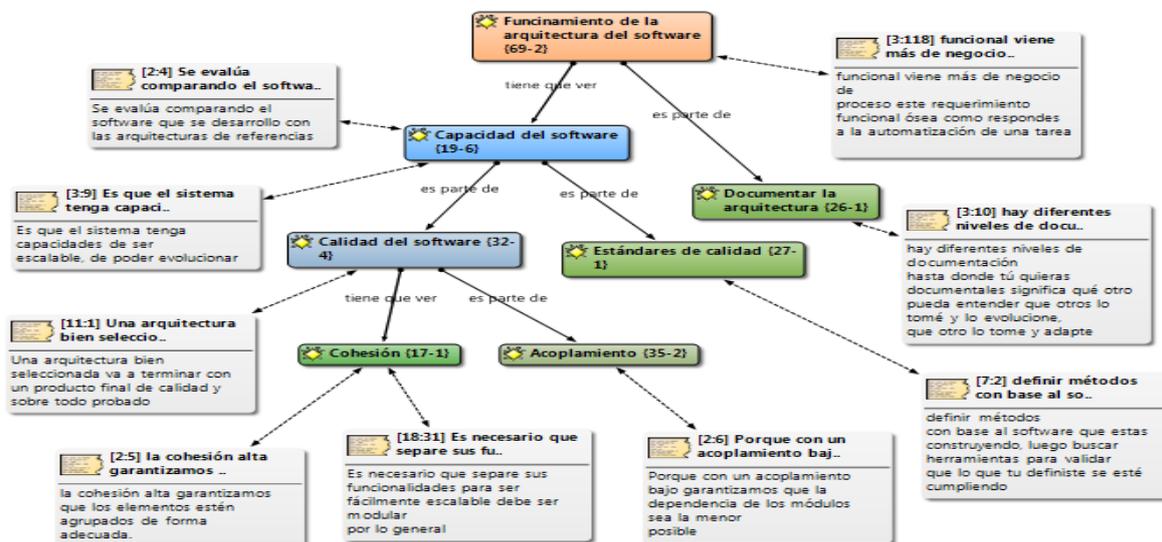


Figura 13. Tercera red semántica: Funcionamiento de la arquitectura de software.

como las necesidades del cliente. Para el (Informante 20, cita 80), *“Con el tema de la documentación, hay que tener mucho cuidado, porque si no se tiene la documentación al día, esto puede crear confusión en lugar de informar”*. La documentación es el registro del cómo se desarrolla o lleva el proyecto. Es el compartir con el equipo la información. *“La documentación facilita el mantenimiento del software”* (Informante 16, cita 29).

Respecto, a los estándares de calidad, la referencia más usada en el Ecuador, es la experiencia y proyectos con éxito. Para el (informante 8, cita 7), *“La calidad del software evalúa la arquitectura, si ha cumplido con todos los patrones o todas las características que necesita tener”*. La definición de la calidad del software, según la IEEE, Std. 610-1990, es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario". IEEE STD 610-1990, IEEE 1990 [10]

El acoplamiento, es la dependencia interna que existe entre las unidades del software. Debe ser lo más bajo posible, es la funcionalidad de una unidad sin depender de otra. Para el (Informante 1, cita 38), *“con un acoplamiento bajo garantizamos que la dependencia de los módulos sea la menor posible, y con la cohesión alta garantizamos que los elementos estén agrupados de forma adecuada.”*

Al diseñar un software, el propósito es que el acoplamiento sea lo más bajo posible. Todo ello, para facilitar la mantenibilidad, detectar problemas y reutilizar las unidades. El acoplamiento, lo define [11] como “El grado de interdependencia entre las unidades de software (módulos, funciones, subrutinas, bibliotecas, etc.) de un sistema informático.”

Con *“la cohesión alta, garantizamos que los elementos estén agrupados de forma adecuada.”* (Informante 1, cita 39). Los

componentes del sistema deben relacionarse, pero no depender uno de otro, para que sea más fácil mantener el sistema. *“la unión de módulos es una relación eficiente, forma un sistema de calidad, interconectados directamente.”* (Informante 16, cita 27).

Es necesario que las funcionalidades estén separadas, para ser fácilmente escalables. Para lo cual, debe ser modular. Por su parte, la cohesión de los módulos del sistema debe funcionar independientemente, con una cohesión alta. La funcionalidad se enfoca a un único propósito, la manera en que se agrupan los datos y los módulos en unidades.

La cuarta red semántica, “Evolución del software”, en ésta resaltan los siguientes códigos: acoplamiento, elección, evolución y modularidad de software, remplazo de versiones y usos de la tecnología. Ver red 4.

Referente a la evolución del software, en el avance de la tecnología, específicamente en el campo de la informática, se presenta el software como un recurso que se emplea en diversos espacios, como respuesta a necesidades que emergen en los sectores industriales, comerciales, financieros, gubernamentales, así como en la salud, educación, artes gráficas, seguridad y de localización, entre otros. La calidad del proceso, al igual que el producto final del desarrollo del software, debe permitir la constante adaptación e incorporación de micro servicios y mantenimiento.

La evolución del software, *“se efectúa contrastando el software, que se desarrolló con las arquitecturas de referencia, para comparar la efectividad y evolución.”* (Informante 1, cita 7). La evolución del software, se evidencia cuando cambian sus partes sin afectar al resto de componentes. Para el (informante 4, cita 2), al referirse a la evolución del software señala que, *“depende en gran parte, del diseño que tuvo al construir el sistema; se debió ya sentar*

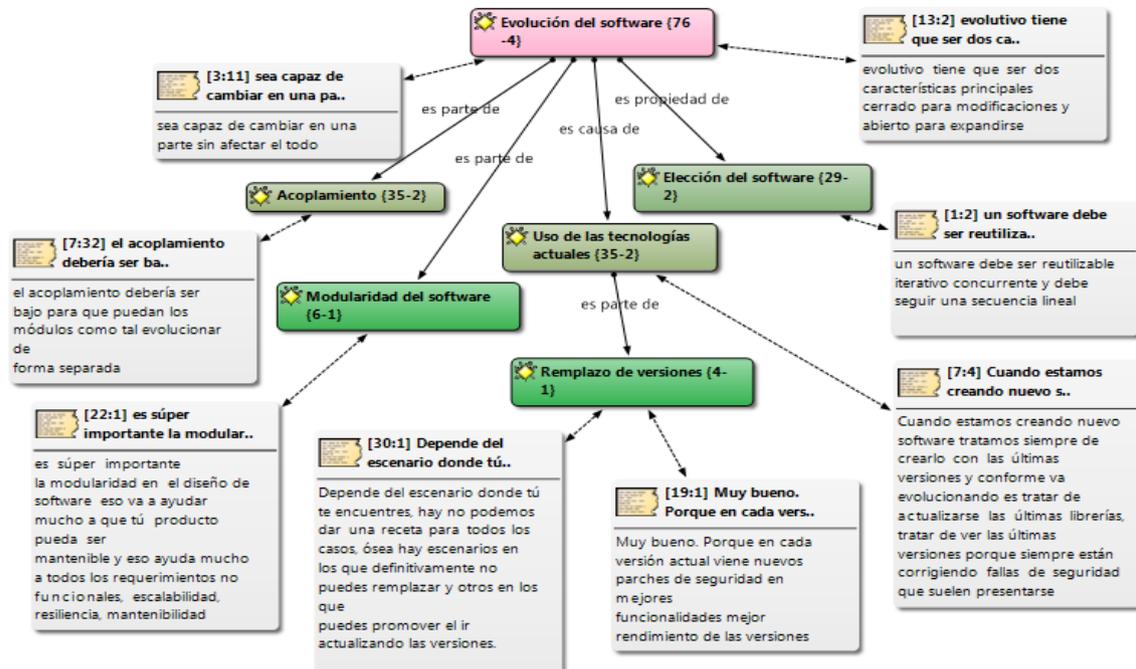


Figura 15. Cuarta red semántica: Evaluación del software.

las bases para que pueda desde cierta manera, el sistema evolucionar”.

Elección del software, “Para escoger una arquitectura, suelo revisar los métodos que están a la vanguardia en las revistas científicas, con la finalidad de optimizar la calidad y no elegir una arquitectura al azar.” (Informante 1, cita 28). De acuerdo al requerimiento del cliente, y las necesidades de servicio, “Básicamente me baso en estándares de las grandes compañías” (Informante 18, cita 6).

La calidad del software, es uno de los aspectos que maneja la programación en la informática; para lo cual, se deben tomar en cuenta los factores externos, que son perceptibles por los usuarios, como el acceso a los programas, facilidad de uso y velocidad. De igual forma, los factores internos que se pueden considerar, como las cualidades, que son detectadas por los profesionales en el diseño del software, tal es el caso de la modularidad y legibilidad del software.

Referente al uso de las nuevas tecnologías, el (Informante 9, cita 12), al referirse al uso de las tecnologías plantea que, “Sí, las uso mucho, para el desarrollo del software en internet, las aplicaciones web me han dado mucha ayuda”. Y para el (informante 6, cita 13) “Básicamente se desarrolla mucho las tecnologías (...) siempre estamos tratando de actualizarnos, en las tecnologías de vanguardia”.

Referente a la quinta red semántica, denominada “Evaluación de la arquitectura de software”, los códigos producto del análisis son: Éxito y efectividad del proyecto, requisitos funcionales y no funcionales, interacción y elección errónea del software. Ver red 5.

La evaluación de la arquitectura de software, según el (Informante 1, cita 7), “se evalúa comparando el software que se desarrolló con las arquitecturas de referencia, para comprobar la efectividad y evolución del software en desarrollo”. La capacidad del software se evalúa, mediante las pruebas que se la aplican al revisar la

posibilidad de crecer en el tiempo, con las pruebas de funcionamiento y de documentación. Respecto a la evaluación del software, el (Informante 10, cita 3) expresa que, “Se deben plantear métricas de evaluación, con ello yo puedo elegir hasta dónde quiero llegar”.

Para el éxito del proyecto, la arquitectura de software es la base del resultado final, “sí una arquitectura es bien seleccionada, va a terminar en un producto final de calidad y sobre todo probado”. (Informante 9, cita 18). Asimismo, el (Informante 19, cita 78), plantea que, “Una buena arquitectura genera un buen producto”. Se considera que la arquitectura bien seleccionada, y diseñada, va a garantizar un producto final de calidad con éxito.

La efectividad del software, determina su funcionalidad y eficacia, esto depende en gran parte de los requerimientos del cliente, para simplificar actividades y resolver conflictos.

De allí, la importancia de clarificar los requerimientos que necesita la empresa. Es

importante, someter el sistema a períodos de pruebas de calidad en el desempeño preliminar, donde participen el diseñador y el cliente. “Yo diría que el éxito de un proyecto en sí, siempre serán los cambios” (Informante 7, cita 9).

Los requisitos funcionales y no funcionales, para el (Informante 1, cita 42), están, en el “Separar las condiciones funcionales de las no funcionales, e identificar los requisitos que se refieren al sistema en conjunto, se debe encontrar un equilibrio adecuado que dependa del tipo de sistema o aplicación que se especifique”.

Referente a la interacción, esta puede ser vista como la acción recíproca que contribuye a la evolución del software, “que tenga una buena experiencia, buena respuesta de usuario” (Informante 13, cita 11).

La elección errónea de la arquitectura de software, se relaciona con la funcionalidad. Esta tiene efectos, sobre los costos que pueden elevarse al igual que el tiempo invertido. “si no inviertes un porcentaje

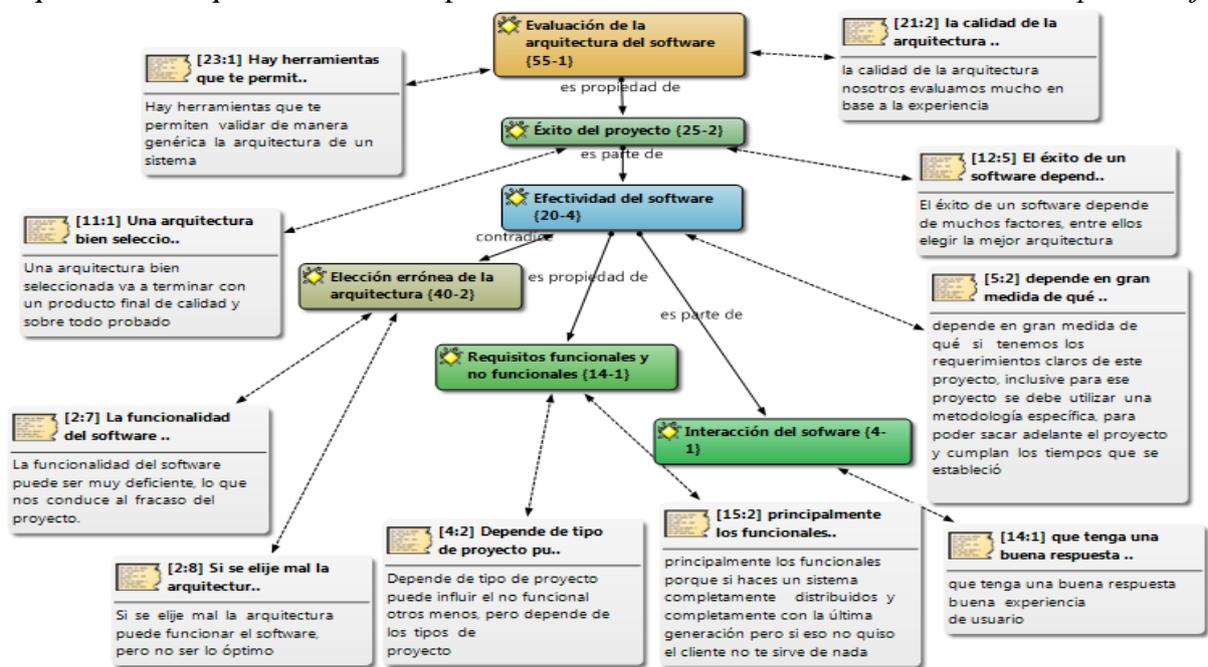


Figura 14. Quinta red semántica: Evaluación de la arquitectura de software.

alto, en diseñar y entender la arquitectura del producto, a largo plazo vas a tener mucho trabajo, porque luego cuando estés ya en el desarrollo, te vas a dar cuenta que no existe, que no tomaste una decisión adecuada” (Informante 19, cita 49).

En la sexta red semántica, que se titula “Arquitectura de software en Ecuador”, los códigos emergentes fueron: actitud y competencias del ingeniero, al igual que la evolución del software en Ecuador. Ver red semántica 6.

Entre los códigos que emergen, están la actitud que manifiestan los ingenieros. Este término lo define [12] como “la postura que se asume, o la respuesta que se manifiesta, ante una determinada situación, en función de los conocimientos del individuo.

Para el (Informante 2, cita 103) refiere que “En el ejercicio de la profesión nos falta mucho”. Esto hace pensar, que en el Ecuador es necesario la actualización

constante del profesional de la informática. Por su parte, el (informante 2, cita 107), expresa que, “lo que sí está mal, es que no tenemos un buen diseño hacia la visión como vamos haciendo el sistema”. Esto ratifica la necesidad de actualización y formación en el campo de la arquitectura de software.

Estas actitudes, evidencian la importancia de la arquitectura para el diseño del software; lo que concuerda con lo expresado por el (Informante 19, cita 28) al señalar que, “En Ecuador, y en general en Latinoamérica, se toma muy en cuenta la arquitectura.” Por su parte, el (Informante 18, cita 4), al referirse al Ecuador plantea que, “por lo general aquí se utiliza bastante el monolito (...) son aplicaciones que todo está ahí mismo y no te permite escalar, de manera más eficiente como los micro servicios”.

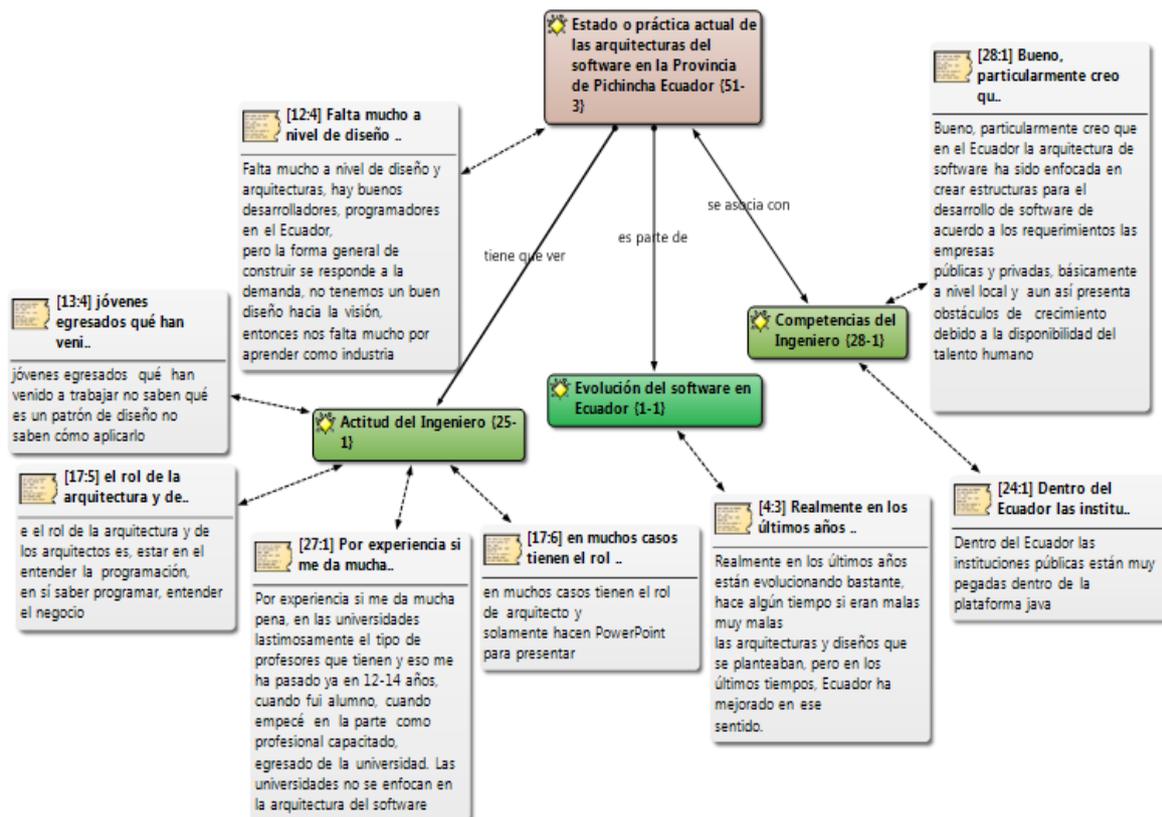


Figura 15: Sexta red semántica: Uso de la arquitectura de software en la provincia de Pichincha.

En el Ecuador, en las instituciones públicas, predominan el uso de la plataforma Java. El cual es un lenguaje de programación. *“Java permite incrustar programas dentro de las páginas Web, para que sean ejecutados en el navegador del usuario”* [13].

En las unidades de análisis, otro código de gran relevancia, refiere a las competencias de los ingenieros en Informática. Al respecto, es importante resaltar la definición del término competencia, por el [14] quien lo define como *“un conjunto de capacidades complejas que le permite a la persona actuar con eficiencia y eficacia, Integra los tres tipos de contenidos: conceptuales (saber), procedimentales (saber hacer) y actitudinales (ser)”*. Referente a las competencias el (Informante 2, cita 16), señala que, *“nos, hace falta bastante en el tema de diseño y arquitectura”*.

Por su parte, las características del desarrollo del software dependen mucho de la comunidad. Para el (Informante 14, cita 1), *“Estas son diferentes a las solicitudes y a las necesidades de la empresa, con su punto de negocio.”* El (Informante 3, cita 28), al referirse al tema del software señala que, *“Realmente en los últimos años se ha avanzado bastante”*.

Para concluir, el análisis sobre los elementos que interviene en el diseño y programación de la arquitectura de software. Este término emergió como la categoría dominante. Las áreas con mayor demanda del software en el Ecuador, son el campo de la tecnología educativa, formación online; el comercio y negocios electrónicos, seguridad y vigilancia, y el sector salud, entre otros.

4.3. Códigos emergentes

Luego de haber desarrollado un análisis cualitativo, en lo referente a la arquitectura que se emplea en la industria de software, en la Provincia de Pichincha, en la

República del Ecuador; se tomó en cuenta, los sectores más representativos, entre los que están el gubernamental, empresarial y universitario, para lo cual, se consultó a los expertos en la materia que laboran en estos espacios, y así saber su posición respecto al impacto y desarrollo de la arquitectura de software, y poder evaluar el estado actual.

Entre los aspectos emergentes, que se produjeron al desarrollar los objetivos de la investigación, están en primera línea y como primer constructo, el uso de la arquitectura de software.

Entre las acciones pertinentes, está el diseño del producto que desarrollan los programadores, para dar respuesta o solución a un problema o requerimiento específico. Es por ello, que se evalúa constantemente la capacidad y calidad de los componentes que integran el software. De allí, la importancia de los métodos y conocimientos que aplican los diseñadores para la elección de la arquitectura.

Según lo emergente en la investigación, muchos de ellos se basan en la experiencia personal del ingeniero, programador o diseñador y los requerimientos funcionales que demanda el cliente.

A continuación, se presenta una tabla resumen, producto del análisis inductivo de la información.

Tabla 12. Categorización de la información.

Fuente	Código	Nº	Subcategoría	Categoría
Entrevista a veinte informantes clave	Diseño del software	28	Arquitectura de software (51)	Estado actual del uso de la arquitectura de software, en la Provincia de Pichincha, Ecuador.
	Calidad del software	32		
	Método para elegir la arquitectura	33		
	Conocimientos del software	21		
	Componentes del software	10		
	Capacidad del software	19		
	Características del software	19		
	Costos del software	10		
	Elección errónea del software	40		
	Participación del cliente	31	Metodologías ágiles (65)	
	Metodología empleada	33		
	Elección del software	29		
	Scrum	03		
	Capacidad del software	19	Funcionamiento de la arquitectura de software (68)	
	Calidad del software	32		
	Cohesión	16		
	Acoplamiento	35		
	Documentar la arquitectura	26		
	Estándares de calidad	27		
	Modularidad	05	Evolución del software (76)	
	Acoplamiento	35		
	Elección del software	29		
	Uso de las tecnologías actuales	35		
Reemplazo de versiones	03			
Éxito del proyecto	25	Evaluación de la arquitectura de software (55)		
Efectividad del software	9			
Requisitos funcionales y no funcionales	14			
Interacción en el software	04			
Elección errónea de la arquitectura	40			
Actitud de ingeniero	24	Arquitectura de software En la provincia de Pichincha (51)		
Competencias del ingeniero	27			
Evolución del software	76			

Para los programadores, la arquitectura de software depende de las características que se esperan del producto, costos de producción y mantenimiento. Una elección errónea del software, podría dar respuesta en el momento, pero en la medida que el software evoluciona, va a requerir de modificaciones y no va a poder soportar nuevos componentes, que fuesen necesario incorporar.

Las metodologías ágiles, se utilizan en la provincia de Pichincha, permiten adaptar el trabajo a las condiciones que exige el proyecto. Se caracterizan por propiciar los requerimientos técnicos, operacionales y flexibles, que los ingenieros adaptan

fácilmente a los cambios, en función de las necesidades que requieren los clientes. Las metodologías ágiles propician el trabajo en equipo, donde el cliente aporta la información necesaria, para el diseño y evolución del software. También, se aplica el scrum como un método de trabajo colaborativo en fases.

Uno de los aspectos, que emergieron con mayor fortaleza, es la importancia que dan los programadores y diseñadores, al diseño y documentación de la arquitectura de software. Estas acciones, se expresan en la calidad y continuidad de la evaluación constante del producto. Otro aspecto, al que se le da gran relevancia es al acoplamiento, entendido como la interdependencia en la funcionalidad del módulo y la cohesión, que es el cómo se relacionan las funciones en el módulo independiente. Es por ello que, la cohesión debe ser alta y el acoplamiento bajo.

4.3 Principales Categorías que Emergieron en la Investigación

En cuanto al uso de la arquitectura de software, en la Provincia de Pichincha, en el Ecuador, entre los factores emergentes están la actitud de los ingenieros y sus competencias, acciones que van a influir en la evolución del software en el país. Es importante resaltar que, en cuanto a las competencias existe consenso, en que falta capacitación, lo cual es una actitud positiva por parte de los profesionales consultados, pues existe disposición para la formación y mejoramiento profesional.

Es por ello, que es necesario el desarrollo de planes y programas de actualización, que permitan a los diseñadores y programadores acceder a los conocimientos informáticos actualizados. Estas acciones, podrían incidir favorablemente a la optimización y evolución de la arquitectura de software, en la provincia de Pichincha, en el Ecuador.

4.4 Teoría Emergente

Entre los constructos que emergieron al analizar el uso de la arquitectura de la industria del software en la provincia de Pichincha, predominó la experiencia y formación del programador, así como el trabajo en equipo con el cliente, para satisfacer sus necesidades.

Tabla 13. Conceptos Emergentes en la Investigación.

	Arquitectura de software	Metodologías ágiles	Funcionamiento de la arquitectura de software	Evolución del software	del	Evaluación de la arquitectura de software	Práctica actual del software en la provincia
Arquitectura	Es la estructura que soporta el sistema.	Son las prácticas empleadas en la arquitectura de software, su objetivo es entregar una solución con un alto grado de calidad en un periodo corto de tiempo.	Una arquitectura abierta al cambio.	La evolución del software se evidencia cuando cambian sus partes sin afectar al resto de componentes.		Se evalúa comparando el software que se desarrolló con las arquitecturas de referencia, para comprobar la efectividad	Las áreas con mayor demanda es la tecnología educativa, formación online; el comercio y negocios electrónicos, seguridad y vigilancia, y el sector salud, entre otros.
Diseño	Es en función del producto esperado y la necesidad que tengan los clientes.	Son las prácticas empleadas en el diseño del software, su objetivo es entregar una solución con un alto grado de calidad en un periodo corto de tiempo.	Debe llenar las expectativas de los usuarios, cubrir sus requerimientos.	La evolución del software se evidencia cuando cambian sus partes sin afectar al resto de componentes.		El diseño y desarrollo se evalúa constantemente, para obtener logros efectivos.	El diseño depende de la experiencia y la referencia.
Calidad	El rendimiento en función de los resultados obtenidos por el cliente.	Propician el desarrollo de proyectos flexibles.	Acoplamiento bajo y la cohesión alta.	Depende en gran parte del diseño empleado al construir el sistema.		Una buena selección va a determinar la calidad del producto final.	La experiencia y la referencia.
Participación del cliente	Para la construcción de la arquitectura, es muy importante la participación del cliente.	Trabajo en equipo diseñador y cliente, se adaptan a los cambios y necesidades de los clientes.	El cliente es una fuente importante de información, basada en las experiencias sobre las necesidades e intereses de la empresa.	Es determinante para saber el tipo de requerimientos necesarios y como pueden evolucionar.		Debe presentar atributos comunes de calidad, rendimiento, seguridad y capacidad de administración.	Al aplicar las metodologías ágiles, se trabaja en equipo integrado por el diseñador y el cliente, para precisar los requerimientos necesarios.

Documentar	Es la diagramación y registro de todos los elementos y estructuras que integran el sistema que componen el software.	Se debe tener mucho cuidado porque si no se tiene la documentación al día puede crear confusión en lugar de informar.	Se registra que tipo de relación e interacción que existe entre los componentes del software.	La documentación es el registro del cómo se desarrolla o lleva el proyecto.	Es el compartir con el equipo la información, que facilita el mantenimiento del software.	Es parte del trabajo del diseñador, registrar para comunicar lo desarrollado, al equipo de trabajo.
Éxito del proyecto	Se basa en la experiencia de los programadores	Conformar un equipo colaborativo de trabajo integrado por los diseñadores del software y la participación del cliente.	Acoplamiento bajo y cohesión alta.	La evolución del software se evidencia cuando cambian sus partes sin afectar los demás componentes.	Se evalúa comparando el software que se desarrolló con las arquitecturas de referencia.	En los últimos años, en el Ecuador se ha avanzado bastante.
Actitud del ingeniero	Esto ratifica la necesidad de actualización y formación en el campo de la arquitectura de software.	Es el método que más se utiliza en la Provincia de Pichincha.	Por lo general aquí se utiliza bastante el monolítico.	Si ha evolucionado de acuerdo a las necesidades del mercado.	Por referencia y experiencia.	Dentro del Ecuador las instituciones públicas están muy pegadas dentro de la plataforma java tienen a crear arquitecturas bastantes monolíticas.
Competencias del ingeniero	En el Ecuador es necesario la actualización constante del profesional de la informática.	Emplean con frecuencia las metodologías ágiles.	Falta bastante capacitación en el tema de la arquitectura de software.	Falta formación, a la par del avance tecnológico.	Por referencia y experiencia.	La arquitectura de software depende de las características que se esperan del producto, costos de producción y mantenimiento.

4. CONCLUSIONES

Una vez concluida la investigación se puede mencionar las siguientes conclusiones: con respecto a la utilización de la arquitectura de software y su estado actual, en la provincia de Pichincha en el análisis cuantitativo:

Existe una clara intención, de establecer una nueva arquitectura de desarrollo de aplicaciones, acorde a la utilización de nuevas tecnologías, que faciliten el trabajo y la estructuración en la etapa de desarrollo. Esto evidentemente, obliga a los profesionales de la informática, a ir a la par con el desarrollo y aparición de nuevas tendencias que día a día irán evolucionando y emergiendo en pro del progreso, de las nuevas tendencias del mercado del software.

El actual desarrollo del software, bajo las metodologías ágiles, gira alrededor de propiciar de forma rápida, soluciones con un alto grado de calidad en el menor tiempo posible y respetando los principios y las restricciones de la arquitectura de software diseñada, permitiendo ser más competitivos para producir softwares eficaces que satisfagan el mercado. Al separar funcionalidades de las aplicaciones, esto representa una ventaja al momento del mantenimiento, cambio o mejoras de los aplicativos.

Es evidente, el auge de las nuevas tecnologías, así como las tendencias que impulsan a los usuarios a estar actualizados. Para los profesionales en el área de la informática, es indudable que, si no están a la par de las nuevas metodologías y nuevos softwares de desarrollo, como nuevas tendencias y necesidades del mercado, corren el riesgo de perder oportunidades.

Con respecto al análisis cualitativo, sobre el mismo tema del estudio podemos indicar lo siguiente:

El diseño y aplicación del producto, depende de la necesidad y cantidad de usuarios; para lo cual, la calidad y capacidad tienen una relación directa con las características y costos. La evaluación, es una acción constante para obtener el éxito del proyecto, en cuanto a funcionalidad y eficacia. Por su parte, la elección y diseño de la arquitectura, garantiza el buen funcionamiento y evolución del software. Se emplean las metodologías ágiles, apoyado en la participación del cliente. De igual manera, la documentación de la arquitectura, se realiza con la finalidad de registrar qué tipo de relación e interacción existe, entre los componentes de la arquitectura del computador, para satisfacer tanto los objetivos de la empresa como las necesidades del cliente. Por su parte, el acoplamiento debe ser a través de la alta cohesión y los métodos para la elección del software, es la experiencia y comparación con métodos de vanguardia, en revistas científicas.

Cabe destacar qué, las instituciones públicas actualmente utilizan la plataforma JAVA. Por su parte, los ingenieros manifestaron actitudes positivas sobre la necesidad de la formación y actualización constante, para superar debilidades que se presentan, y así mejorar las competencias en cuanto al diseño y aplicación de la arquitectura de software.

5. RECOMENDACIONES

Este trabajo permitió, dar opiniones diversas con respecto a la investigación, es por ello, que es importante, poder mencionar que las Universidades de manera general, pueden establecer convenios para el desarrollo de investigaciones en empresas de toda índole. Esto principalmente, debido a las dificultades encontradas para coordinar las entrevistas en diversas empresas, ya que muchas de

ellas ofrecieron resistencia y poca colaboración por parte de las mismas, para lograr obtener la opinión profesional de su personal.

6. REFERENCIAS

- [1] S. Martínez, S. Arango y J. Robledo, «El Crecimiento de La Industria del Software en Colombia: Un Análisis Sistémico,» *revista@eia.edu.co*, 2015.
- [2] M. Espinoza y D. d. P. Gallegos, «La industria del software en Ecuador: evolución y situación actual,» *Revista EspacioS*, 2017.
- [3] L. Veliz Rojas y A. Bianchetti Saavedra, «Integración Metodológica en la Investigación de Fenómenos Complejos en Enfermería,» Universidad de La Serena, Coquimbo - Chile, 2016.
- [4] M. Á. Cea D'Ancona, *Metodología Cuantitativa Estrategias y Técnicas de Investigación social*, Madrid: Síntesis S.A., 1996.
- [5] Explorable, «Investigación Cuantitativa y Cualitativa,» Explorable.com, 2008-2019. [En línea]. Available: <https://explorable.com/es/investigacion-cuantitativa-y-cualitativa>. [Último acceso: 20 07 2019].
- [6] H. Sarasty, «Documentación y Análisis de los Principales Frameworks de Arquitectura de Software en Aplicaciones Empresariales,» 10 octubre 2015. [En línea]. Available: http://sedici.unlp.edu.ar/bitstream/handle/10915/52183/Documento_completo..pdf?sequence=3.
- [7] M. Navarro, M. Moreno, J. Areanda, L. Parra, J. Rueda y J. Pantano, «Integración de Arquitectura de Software en el Ciclo de Vida de las Metodologías Ágiles,» *Repositorio Institucional de la UNLP*, 2017.
- [8] Y. Amaya, «Metodologías Ágiles en el Desarrollo de Aplicaciones,» *Journal Technology, Volumen 12, N°2,* pp. file:///C:/Users/adminn/Downloads/Dialnet-Metodologias Agiles En El Desarrollo DeAplicaciones Para-6041502.pdf, 2013.
- [9] A. Reyes, A. González y M. André, «Modelo Básico Inicial de Calidad Externa para Productos de Software,» *Revista Cubana de Ciencias Informáticas*, 2016
- [10] Universidad del Cauca, «Calidad del Software, Programa de Ingeniería de Sistemas,» 2017. [En línea]. Available: http://artemisa.unicauca.edu.co/~cardila/CS_01_Conceptos_Iniciales_2017.pdf.
- [11] ALEGSA, «Diccionario de Informática y Tecnología,» 9 Septiembre 2019. [En línea]. Available: <http://www.alegsa.com.ar/Dic/acoplamiento.php>.
- [12] Diccionario de Psicología, «Plethora,» 30 septiembre 2019. [En línea]. Available: <http://consulta-psicologica.com/diccionario-de-psicologia/32-actitud.html>.
- [13] P. Sznajdleder, *Java a Fondo, Estudio de Lenguaje de Aplicaciones*, Buenos Aires, Argentina: Alfaomega Grupo Editor Argentino, 2013.
- [14] «Glosario Pedagogía,» 25 mayo 2019. [En línea]. Available: https://glosarios.servidor-alicante.com/pedagogia#terminos_C.
- [15] A. M. Fúquene y O. F. Castellanos, «Basis for Implementing a Model of

Intelligence for Strengthening the
Technological Development of the
Software Industry and its Associated
Services in Colombia,» 2007.