



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

TÍTULO:

**Estudio sobre el impacto de la administración de proyectos en
las empresas de desarrollo de software en la ciudad de
Guayaquil**

AUTORA:

Salgado Ontaneda Rosemarie

**Trabajo de Titulación previo a la obtención del Título de
INGENIERO EN SISTEMAS COMPUTACIONALES**

TUTORA:

Ing. Adela María Eugenia Zurita Fabre, Mgs

**Guayaquil, Ecuador
2014**



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERIA
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

CERTIFICACIÓN

Certificamos que el presente trabajo fue realizado en su totalidad por **Rosemarie Salgado Ontaneda**, como requerimiento parcial para la obtención del Título de **Ingeniero en Sistemas Computacionales**.

TUTORA

Ing. Adela Zurita Fabre, Mgs

REVISORES

Ing. Lenín Freire Cobo, Mgs

Lcda. Ruth Zambrano Saltos, Mgs

DIRECTORA (e) DE LA CARRERA

Ing. Beatriz Guerrero Yépez, Mgs

Guayaquil, a los 21 del mes de marzo de 2014



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERIA
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

DECLARACIÓN DE RESPONSABILIDAD

Yo, Rosemarie Salgado Ontaneda

DECLARO QUE:

El Trabajo de Titulación **Estudio sobre el impacto de la administración de proyectos en las empresas de desarrollo de software en la ciudad de Guayaquil** previa a la obtención del Título de **Ingeniero en Sistemas Computacionales**, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros conforme las citas que constan en las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance científico del Trabajo de Titulación referido.

Guayaquil, a los 21 del mes de marzo de 2014

LA AUTORA

Rosemarie Salgado Ontaneda



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERIA
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

AUTORIZACIÓN

Yo, **Rosemarie Salgado Ontaneda**

Autorizo a la Universidad Católica de Santiago de Guayaquil, la **publicación** en la biblioteca de la institución del Trabajo de Titulación: **Estudio sobre el impacto de la administración de proyectos en las empresas de desarrollo de software en la ciudad de Guayaquil**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 21 del mes de marzo de 2014

LA AUTORA:

Rosemarie Salgado Ontaneda

AGRADECIMIENTO

Un reconocimiento especial a la Universidad Católica de Santiago de Guayaquil, institución de Educación Superior de alto reconocimiento académico, por haberme brindado la oportunidad de cursar esta noble carrera y llegar al esperado término.

A la Carrera de Ingeniería en Sistemas Computacionales y sus docentes, quienes supieron apreciar la calidad del trabajo realizado, no sólo en esta etapa final de la vida estudiantil, sino durante toda la carrera; docentes que comprendieron las dificultades que significó para mí el retorno a las aulas de clase luego de un largo período de ausencia. Mi reconocimiento a la Ing. Adela Zurita Fabre, tutora de Trabajo de Titulación, a los (as) Ingenieros (as), Lenín Freire Cobo, Lector de Contenido, Inelda Martillo Alcívar, Darwin Cercado Barragán, Galo Cornejo Gómez, Edison Toala Quimí.

A todos ellos, muchas gracias.

ROSEMARIE SALGADO ONTANEDA

DEDICATORIA

La investigación que condujo a la culminación de este Trabajo de Titulación se hizo realidad gracias a la ayuda y colaboración de varias personas, que de una u otra forma, directa o indirectamente, fueron partícipes en la consecución de este objetivo.

Para mi recordado hermano Billy Salgado Ontaneda que, aunque no tuvo la satisfacción de ver culminados con éxito mis estudios universitarios, fue la persona que supo orientarme en la nueva vida que debía emprender, en la instrucción académica que me hacía falta; para él, siempre fue primordial la preparación universitaria.

En el ámbito laboral, para el Ing. José Paredes Espinel, por su apoyo incondicional en este período de Titulación, al permitir y facilitar el desarrollo de la investigación en horas de poca confluencia de trabajo.

Para los compañeros con los que se compartió momentos de premura, de satisfacción, de desaliento, de gozo en el diario convivir universitario, en especial para el hoy ingeniero Washington Yépez Yépez, por ser una de las pocas personas que tuvo la paciencia necesaria para atender las constantes dudas e inquietudes académicas por mí requeridas, por colaborar conmigo cuando fue necesario durante el transcurso de la carrera, por estar siempre presente, por ofrecer su apoyo cuando todo parecía perdido. Compañeros de aula que, con su pequeño grano de arena y gran voluntad de su parte, demostraron solidaridad en momentos apremiantes en el desarrollo de este proyecto, entre ellos a Sully Flores Bolaño, Michelle Herrera Boza, Teddy Arteaga Zambrano, Jaime Lema Loor, Roy Romero Coloma.

Por último, para todos aquellos que apoyaron y colaboraron desinteresadamente mientras se llevó a cabo el estudio.

ROSEMARIE SALGADO ONTANEDA

TRIBUNAL DE SUSTENTACIÓN

Ing. Adela Zurita Fabre, Mgs
PROFESORA TUTORA

Ing. Lenín Freire Cobo, Mgs
PROFESORA DELEGADA

Lic. Ruth Zambrano Saltos, Mgs
PROFESORA DELEGADA



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE INGENIERIA
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

CALIFICACIÓN

**Ing. Adela Zurita Fabre, Mgs
PROFESORA TUTORA**

ÍNDICE GENERAL

| | |
|--|------|
| ÍNDICE DE TABLAS | xii |
| ÍNDICE DE GRÁFICOS | xiii |
| ÍNDICE DE IMÁGENES..... | xiv |
| ÍNDICE DE ANEXOS..... | xv |
| RESUMEN..... | xvi |
| ABSTRACT..... | xvii |
| | |
| CAPITULO I: ANTECEDENTES DE LA INVESTIGACIÓN..... | 10 |
| | |
| 1.1 Marco Teórico | 10 |
| 1.1.1 Metodologías de desarrollo | 11 |
| 1.1.1.1. Modelo Lineal Secuencial (Cascada)..... | 11 |
| 1.1.1.2. Modelo de Construcción de Prototipos..... | 14 |
| 1.1.1.3. Modelo DRA (Desarrollo Rápido de Aplicaciones)..... | 17 |
| 1.1.1.4. Modelos evolutivos del proceso del software | 19 |
| a) Modelo Incremental..... | 19 |
| b) Modelo Espiral | 21 |
| 1.1.1.5. Metodologías ágiles | 23 |
| a) Programación extrema..... | 24 |
| b) SCRUM..... | 25 |
| 1.1.1.6. RUP..... | 26 |
| 1.1.2 Pruebas de software..... | 28 |
| 1.1.2.1 Prueba de caja blanca..... | 32 |
| 1.1.2.2 Prueba de caja negra | 34 |
| 1.1.2.3 Prueba de la ruta básica..... | 36 |
| 1.1.2.4 Prueba de la estructura de control..... | 38 |
| 1.1.3 Métricas de software | 40 |
| 1.1.3.1 Métricas para el modelo de análisis | 42 |
| 1.1.3.2 Métricas para el modelo de diseño..... | 44 |
| 1.1.3.3 Métricas para el código fuente..... | 45 |
| 1.1.3.4 Métricas para las pruebas | 46 |
| 1.1.4 Gestión de proyectos..... | 46 |
| 1.1.4.1 El personal..... | 47 |

| | | |
|---|---|-----|
| 1.1.4.2 | El producto | 50 |
| 1.1.4.3 | El proceso..... | 52 |
| 1.1.4.4 | El proyecto..... | 54 |
| 1.1.5 | Gestión de cambios..... | 57 |
| 1.1.5.1 | Control de versiones..... | 59 |
| CAPÍTULO II: MERCADO LOCAL DE SOFTWARE..... | | 61 |
| 2.1 | ¿Qué es la AESOFT? | 61 |
| 2.1.1 | Misión | 61 |
| 2.1.2 | Visión..... | 62 |
| 2.1.3 | Objetivos..... | 62 |
| 2.1.4 | Importancia de la industria..... | 62 |
| 2.1.5 | Datos estadísticos | 63 |
| 2.1.6 | Políticas de la AESOFT | 63 |
| 2.2 | Código de ética de la AESOFT | 64 |
| 2.3 | Repercusión del uso de las tecnologías de la información y la comunicación Tic's en el desarrollo de la economía nacional. | 64 |
| 2.4 | Mercado de software en Ecuador | 65 |
| 2.5 | Ingresos por venta de software en Ecuador..... | 66 |
| 2.6 | Listado de empresas de desarrollo de software según la AESOFT | 89 |
| 2.7 | Descripción de las empresas asociadas a la AESOFT | 91 |
| CAPÍTULO III: METODOLOGÍA | | 106 |
| 3.1 | Enfoque metodológico | 106 |
| 3.2 | Diseño de investigación | 114 |
| 3.3 | Objeto de estudio | 117 |
| 3.4 | Población y muestra..... | 117 |
| 3.5 | Técnicas e instrumentos de recolección de datos..... | 119 |

| | | |
|---|---|-----|
| 3.6 | Técnicas de procesamiento y análisis de datos | 121 |
| CAPÍTULO IV: PRESENTACIÓN DE RESULTADOS | | 122 |
| 4.1 | Prueba piloto aplicada a empresas de desarrollo de software | 122 |
| 4.2 | Análisis de resultados | 122 |
| 4.3. | Entrevista final a empresas de desarrollo de software | 142 |
| 4.4. | Análisis de resultados | 143 |
| 4.5. | Entrevista para medir la satisfacción del usuario | 165 |
| 4.6. | Análisis de resultados | 165 |
| CAPÍTULO V: INFORME FINAL | | 176 |
| CONCLUSIONES Y RECOMENDACIONES | | 178 |
| REFERENCIAS BIBLIOGRÁFICAS..... | | 181 |
| GLOSARIO | | 183 |
| ANEXOS | | 184 |

ÍNDICE DE TABLAS

| | |
|--|----|
| Tabla 1: Ventajas y desventajas del Modelo Cascada..... | 12 |
| Tabla 2: Ventajas y desventajas del modelo de Construcción de Prototipos.... | 17 |
| Tabla 3: Ventajas y desventajas del modelo Incremental | 20 |
| Tabla 4: Exportaciones de software en dólares | 88 |

ÍNDICE DE GRÁFICOS

| | |
|--|-----|
| Gráfico 1: Modelo Cascada | 13 |
| Gráfico 2: Entrega incremental | 21 |
| Gráfico 3: Casos de Prueba..... | 32 |
| Gráfico 4: Tipos de Bucles | 40 |
| Gráfico 5: Enfoque cuantitativo de la investigación..... | 109 |
| Gráfico 6: Fases del proceso cuantitativo | 110 |
| Gráfico 7: Enfoque cualitativo de la investigación..... | 112 |
| Gráfico 8: Fases del proceso cualitativo | 113 |

ÍNDICE DE IMÁGENES

| | |
|---|----|
| Imagen 1: Modelo Espiral | 22 |
| Imagen 2: Prueba de Caja Blanca | 33 |
| Imagen 3: Prueba de Caja Negra | 36 |
| Imagen 4: Prueba de la Ruta Básica | 37 |
| Imagen 5: Listado de empresas asociadas a la AESOFT | 90 |

ÍNDICE DE ANEXOS

| | |
|---|-----|
| Anexo 1: Código de Ética de la AESOFT | 184 |
| Anexo 2: Listado de empresas proporcionado por la AESOFT..... | 189 |
| Anexo 3: Formato de entrevista..... | 190 |
| Anexo 4: Formato final de entrevista | 194 |
| Anexo 5: Entrevista para medir la satisfacción del usuario..... | 199 |

RESUMEN

Comprender la realidad de las empresas dedicadas al desarrollo y producción de software de calidad es necesario para determinar qué tipo de sociedad se encuentra entre las preferencias de los clientes que requieren de este servicio. Empresas locales del mercado de software son el escenario para indagar sobre su funcionamiento interno: tipo de empresa, años de funcionamiento, cantidad de proyectos realizados y en curso, metodologías de desarrollo para la producción del software, problemas al recibir los requerimientos, métricas para la evaluación del software, el porcentaje de errores en la producción del mismo, la pronta atención a los requerimientos de los usuarios, son aspectos que influyen en los clientes que se deciden por una determinada solución. La muestra tomada de la AESOFT y otras empresas dedicadas a esta actividad, luego de aplicada la entrevista, permitió conocer los resultados en la aplicación de las metodologías de desarrollo de proyectos de software que utilizan las empresas de software de la ciudad de Guayaquil, concluyéndose que la industria del software es relativamente joven, dirigida por jefes/directores hombres, en su totalidad hasta el momento, utilizando metodología de desarrollo tradicional con un despunte de los métodos ágiles y con una preferencia del usuario, en su mayoría, por recomendación de otras personas. Se plantearon recomendaciones a ser tomadas en cuenta por las empresas de desarrollo que brindan el servicio para verificar la efectividad del software.

Palabras Claves: metodología, desarrollo, proceso, software, espiral, cascada

ABSTRACT

Understanding the reality of companies dedicated to the development and production of quality software it's necessary to determine what type of society is found between the clients' preferences that require this service. Software market's local companies are the scenery to inquire into their internal functioning: company type, functioning years, quantity of achieved projects and still under development, development methodologies for software production, quick attention of the users' requirements, are aspects that influence on the clients that decide for a determined solution. The sample taken from AESOFT and other companies dedicated to this activity, after being applied the interview, it allowed to be known the results in the application of development methodologies of software projects that software companies from Guayaquil's city use, concluding that software industries are relatively young, directed by men bosses/directors, so far in its totality, using a traditional development methodology with a blunt of agile methods and user's preference, in its whole, by another person's recommendation. Recommendations were proposed to be taken into account by development companies that share this service to verify the software's effectiveness.

Key Words: Methodology, development, process, software, spiral, cascade

INTRODUCCIÓN

La globalización, fenómeno económico, tecnológico, social y cultural a gran escala, obliga a las compañías a incrementar su competitividad. Las grandes y pequeñas empresas han ido asumiendo un prominente rol en ofrecer servicios tecnológicos, oportunidades de inversión e innovación, y por lo tanto, se han ido convirtiendo paulatinamente en la columna vertebral de muchos profesionales que ven en la gestión de proyectos de tecnología una puerta hacia la generación de puestos de trabajo y por ende, generación de ingresos.

Si una empresa dedica sus actividades a desarrollar software debe tener presente el verdadero concepto de su negocio, esto es, su actividad es un negocio de software, y la adopción de un modelo determinado para la gestión de proyectos, desarrollo y mantenimiento de sistemas a implementar pueden presentar inconvenientes notables que influyan en la eficiencia al gestionar el negocio. Los empresarios que pretendan utilizar métodos de mayor eficiencia para la gestión de sus actividades, pueden incurrir en la incertidumbre ante la gama de modelos de desarrollo de calidad, procesos definidos y técnicas de trabajo que han aparecido en los últimos años; o, en su defecto, abarcar al primer modelo que aparece a los pies de su organización, adoptándolo como “una solución” de eficiencia, eficacia y calidad en la gestión de proyectos. La intención de este estudio es presentar la situación general de las empresas de software como una orientación para directores técnicos, responsables de departamentos informáticos y personal directivo de organizaciones de software para que analicen en su totalidad la problemática de que el entorno en el cual se desenvuelven es un ambiente sistémico, en donde los componentes de toda la organización se deben acoplar y trabajar en forma conjunta, a la par de las características, cultura y estrategias de la empresa.

A partir de un estudio realizado por la Escuela Superior Politécnica del Litoral (ESPOL) referente a un estudio estadístico exploratorio de las empresas desarrolladoras de software asentadas en las tres principales ciudades del Ecuador Guayaquil, Quito y Cuenca (Salazar, Villavicencio, Macías, & Snoeck, 2004), con el fin de conocer las características más importantes de las empresas desarrolladoras de software, se tiene previsto realizar un estudio que abarque un análisis sobre una muestra del universo de la industria del software en la ciudad de Guayaquil considerada como una de las ciudades más pobladas en las que se concentra un gran movimiento económico en el país.

La importancia de este estudio se deriva en que actualmente, se ha presentado la necesidad de que las empresas y organizaciones de distinta índole se integren a los nuevos sistemas de información para que puedan participar de todos los avances de tecnología disponibles. Esto ha obligado a las empresas a tratar de encontrar soluciones tecnológicas en base a la gestión de proyectos de tecnología, para brindar a los usuarios servicios de calidad tomando en cuenta normas y directrices establecidas que las coloquen a la vanguardia de la eficacia y eficiencia de los servicios que prestan.

Relacionado con lo anterior, es necesario mencionar que los requisitos de software se han convertido en un punto clave en el desarrollo de las aplicaciones informáticas. Muchos proyectos de desarrollo de software se pierden por una mala definición, interpretación, especificación o administración de los requisitos. Si existen requisitos incompletos, cambios mal efectuados o administrados pueden conducir al fracaso contundente de un proyecto de software (Universidad Autónoma de Sinaloa, s.f.). Lo mencionado afectaría a la calidad de los productos de software ofertados por las empresas de desarrollo y, por ende, marcará la preferencia y permanencia o no de una empresa de software en el competitivo mercado tecnológico, por lo que la búsqueda constante de mejora en la formación de sus profesionales y las estrategias de

marketing que se empleen, serán las directrices a seguir por los empresarios sobre las cuales se sienten los cimientos de la empresa.

La importancia descrita hace que se resalte el valor de este estudio, desde puntos de vista tanto científico como social. En lo científico, el realizar un estudio de carácter investigativo que dé a conocer las características más relevantes de la realidad de las empresas de desarrollo de software asentadas en la ciudad de Guayaquil, permitirá determinar las necesidades informáticas actuales del mercado local. Se debe destacar la necesidad de aplicar modelos de procesos y de calidad de los mismos, con el fin de brindar eficiencia y consistencia en los resultados de los proyectos, para sentar las bases para una mejora continua conforme a las nuevas metodologías implementadas y que servirán de modelo a seguir.

La industria del software es una industria que trabaja con un conocimiento profesional que se encuentra escasamente asentado. Por ende, la aparición de modelos de calidad se aumenta al movimiento paralelo de propuestas de currículos profesionales.

En cuanto al punto de vista social, a través de este estudio se busca recabar información de las empresas de desarrollo de software, en cuanto se refiere a la formación de sus profesionales y su desarrollo profesional continuo; se buscaría además ofrecer retroalimentación a las universidades y beneficiar tanto a estudiantes como al mercado laboral:

A las *universidades*, porque permitiría la revisión de los pensum académicos de sus ofertas académicas para que, con nuevos contenidos y metodologías acordes a los cambios y adelantos tecnológicos, puedan acoplarse a los desafíos que presenta la administración de la información, ofreciendo un abanico de nuevas posibilidades de especialización y práctica de la futura profesión.

Al *estudiante*, porque a través de formación especializada, adquiera una gama de nuevos conocimientos a través de la práctica. De este modo, el perfil del propio estudiante se complementa dando paso a profesionales de excelente calidad capaces de hacer frente a los retos de su profesión y permitiendo ocupar sitios cotizados en las organizaciones.

Al *mercado laboral*, porque los profesionales que surgen de las aulas de clases encuentran un nicho de mercado en el que se pueden insertar fácil y rápidamente y ofrecer todo su contingente de conocimientos actualizados y experiencia para mejorar los resultados de las empresas de este sector.

Con el fin de identificar la problemática que motivó este estudio, se debe orientar la mirada hacia las empresas que desarrollan software y a los usuarios de sus servicios, y destacar que el software es la herramienta que apoya a empresarios en la toma de decisiones oportuna y efectiva para su permanencia en el mercado, es también un fuerte pilar para la investigación científica moderna y de resolución de problemas de ingeniería, ya que está inmerso en sistemas de telecomunicaciones, procesos industriales, entretenimiento, entre otros. Conforme transcurre el tiempo, el desarrollo del software contribuye a la realización de los grandes cambios que se presentan en la actualidad, desde lo más básico como la educación elemental hasta lo más avanzado, como la ingeniería genética. Por todo esto es que, manejar proyectos en el área informática, cobra cada día mayor importancia. Entre las nuevas necesidades de las empresas se encuentra la de “optimizar el uso de los recursos disponibles en una organización para lograr sus objetivos mediante la administración de un método de administración de proyectos integral” (Rivera Martínez & Hernández Chávez, 2010).

Al profundizar en el tema de las actividades de desarrollo de software y, conforme los equipos informáticos se popularizan y bajan su costo, se pone en evidencia nuevos criterios para medir el éxito del desarrollo de software, que se

mantienen hasta la actualidad. Un proyecto es de renombre si el código que se desarrolla tiene (Zavala Ruiz, 2004, pág. 11):

- Responsabilidad adquirida con el proyecto para determinar aspectos de fondo como calendario de inicio y entrega, estimaciones de costos y objetivo de ejecución técnica
- Retroalimentación constante de la empresa que garantiza el proyecto en cuestión
- Retroalimentación constante del usuario que utiliza el sistema
- Responsabilidad del usuario, de quien garantiza el proyecto, involucrado en la determinación de calendarios, estimaciones de costo y objetivo de ejecución técnica
- Organización debidamente estructurada en concordancia al grupo de trabajo del proyecto
- Colaboración del equipo de trabajo que interviene en el proyecto para establecer los calendarios y estimación de costos
- Empeño de quien garantiza el proyecto por crear aptitudes internas
- Métodos de control, sobre todo en referencia a los cambios
- Inteligencia en la utilización de técnicas y procesos de programación en red
- Reducida participación de agencias públicas y gubernamentales

Las tecnologías de la información representan un papel protagónico en la época actual; son la base de la economía de muchos países desarrollados y son la única forma para competir en el mercado moderno incorporándolas a las economías de los países en vías de desarrollo. En base a estos antecedentes, se busca realizar este estudio sobre la gestión de proyectos en las empresas de desarrollo de software asentadas en la ciudad de Guayaquil registradas en la Asociación Ecuatoriana de Software (AESOFT) y otras empresas que están dedicadas a esta actividad, a fin de conocer aspectos relacionados a las

prácticas claves de los procesos que en ellas se ejecutan en cuanto se refiere al manejo de un proyecto. Las fases de un proyecto, la metodología de la gestión de proyectos utilizados en dichas empresas, permitirá develar la realidad de las mismas en el competitivo mercado de las tecnologías de la información (AESOFT, 2011).

En la actualidad, el software cumple una doble funcionalidad: es un producto y al mismo tiempo representa el medio para hacer entrega de un producto o servicio. Como *producto*, entrega las características informáticas del hardware, considerándose como el medio que transforma la información: produce, gestiona, adquiere, modifica, muestra o transfiere información, la misma que puede ser tan simple como lo es un solo bit o tan complejo como una simulación multimedia. Como *medio* para la entrega de un producto, el software es la base de control del equipo informático, la comunicación y transmisión de la información (redes), y el desarrollo y control de otros programas.

Así, con lo anteriormente explicado, cabe definir la hipótesis de esta investigación: *a medida que la Ingeniería de Software se fue tecnificando, las metodologías de producto y de proyectos de software se han ido desarrollando, logrando que la industria del software vaya en pleno crecimiento.* Las variables presentes en esta hipótesis son:

Variable independiente: disponibilidad de nuevas tecnologías y modelos de desarrollo de procesos, que han permitido la tecnificación en el desarrollo de software

Variable dependiente: disponibilidad de recursos y herramientas para el desarrollo de los proyectos.

Planteada la hipótesis e identificadas sus variables, se puede definir los objetivos a lograr con el estudio propuesto. Como **objetivo general**, *Investigar*

los resultados y la aplicación de las metodologías de desarrollo de proyectos de software que utilizan las empresas de software de la ciudad de Guayaquil; para ello se han definido los siguientes **objetivos específicos** que secuencialmente permitirán alcanzar lo enunciado:

- Elaborar un diagnóstico sobre la existencia de empresas que desarrollan software en la ciudad de Guayaquil, su tamaño y antigüedad, las herramientas tecnológicas para prestar el servicio
- Determinar el efecto alcanzado a través de los medios y herramientas aplicables al control de gestión de proyectos orientados al desarrollo de software
- Determinar el nivel de confianza percibida a nivel de usuarios respecto a las empresas que desarrollan software en la ciudad de Guayaquil
- Presentar recomendaciones válidas y aplicables que pudieran ser puestas en consideración para comprobar la efectividad del software, utilizable tanto para las empresas que lo desarrollan así como para las que reciben este servicio

Para establecer una guía en el desarrollo de este estudio, se han definido las siguientes **preguntas de investigación**:

1. ¿En qué consistió el diagnóstico que se realizó sobre las empresas de desarrollo de software de Guayaquil?
2. ¿Se puede determinar si los medios y herramientas para el desarrollo de software ha alcanzado un efecto positivo en la aplicación de la gestión de proyectos?
3. ¿Existe un marcado nivel de confianza en las personas usuarias de las empresas de desarrollo de software de Guayaquil?

4. ¿Qué recomendaciones se pueden poner a consideración tanto para las empresas desarrolladoras como para las personas que usan este servicio?

El estudio planteado se desarrollará con base en dos enfoques metodológicos: el cuantitativo y el cualitativo, lo que se define como enfoque mixto.

Tiene *enfoque cuantitativo* porque está presente un conjunto de procesos. Es secuencial y probatorio; parte de una idea que, luego de delimitarse, se derivan los objetivos y preguntas de investigación; se examina la literatura referente al tema de investigación y se construye un marco teórico. Las preguntas de investigación conducen a la formulación de una o varias hipótesis y se establecen variables; se plantea la comprobación de las hipótesis; se exploran los resultados obtenidos mediante métodos estadísticos y se determinan conclusiones a la o las hipótesis (Hernández Sampieri, Fernández Collado, & Baptista Lucio, 2010, pág. 4)

Es *cualitativo* porque, según Hernández Sampieri et al. (2010, pág. 7), se apoya en la recolección de datos sin medición numérica específica, a manera de descubrir o afinar las preguntas de investigación dentro del proceso de interpretación, es decir, tratando de entender los hechos desde una mirada de los actores que intervienen en el problema de investigación, en otras palabras, cómo dichos autores conciben el problema o situación específica.

La necesidad que surge del problema motivo de la investigación, se lo tratará desde el punto de vista de ambos enfoques: cuantitativo y cualitativo. Las acciones que derivan de estos enfoques estarán dadas en función de la aplicación de los métodos y técnicas de producción de datos, tanto para el *cuantitativo* (entrevista cerrada, análisis cuantitativo de datos secundarios,

cuestionarios), como para el *cualitativo* (recolección de información) aplicados a las empresas de desarrollo de software de la ciudad de Guayaquil.

Este estudio consta de cinco secciones. En la primera se presenta el marco teórico, conceptual del campo de desarrollo de proyectos de software, en la segunda se ofrece un acercamiento directo hacia la industria propiamente dicha del desarrollo de software, la institución que engloba el funcionamiento de empresas que prestan dicho servicio y el funcionamiento interno, algo general, de dichas empresas; en la tercera sección está la metodología seguida para realizar la investigación que sustenta el trabajo de titulación; en una cuarta parte se reúne todos los elementos obtenidos a través de entrevistas, observación directa, encuestas, con su respectivo informe; finalmente en una quinta sección se aprecia las conclusiones y recomendaciones que respaldan el cumplimiento de los objetivos general y específicos determinados inicialmente en este estudio.

CAPITULO I: ANTECEDENTES DE LA INVESTIGACIÓN

1.1 Marco Teórico

El software ha sido y es una de las características primordiales que deben tener todas las organizaciones para poder competir dentro del mercado dedicado a brindar servicios informáticos. Pero antes de adentrarse en la investigación materia de este estudio, es conveniente marcar el punto de partida del mismo y señalar de qué área del conocimiento provienen los conceptos que son medulares para comprender el porqué de la investigación.

Como parte importante del estudio de las ciencias informáticas, se encuentra la Ingeniería de Software que, de acuerdo a Sommerville (2004, pág. 6) “es una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento después que se utiliza”.

Otra definición señala que “la Ingeniería de Software es la disciplina o área de la informática que ofrece métodos y técnicas para desarrollar y mantener software de calidad” (Cabrera, Solano, & Montalván , 2009, pág. 2).

Por lo tanto, las definiciones señaladas en los párrafos anteriores coinciden en que la Ingeniería de Software es una ciencia, disciplina o área encargada de realizar la planificación de todo lo relacionado con el desarrollo del software, desde sus inicios, su puesta en producción y su mantenimiento en las etapas finales.

La Ingeniería de Software tiene como base el proceso. Según Pressman (2005, pág. 24) “el proceso del software forma la base para el control de la gestión de los proyectos de software y establece el contexto en el cual se aplican los métodos técnicos, se generan los productos del trabajo (...), se

establecen los fundamentos, se asegura la calidad y el cambio se maneja de manera apropiada”.

Proceso del software, para Sommerville (2004, pág. 7), es “un conjunto de actividades y resultados asociados que producen un producto de software”.

En definitiva, un proceso de software lo constituyen las técnicas y métodos utilizados en la elaboración de un producto de software, las que son necesarias para garantizar la calidad del mismo.

1.1.1 Metodologías de desarrollo

Existen algunos modelos de procesos de desarrollo. Los más comúnmente conocidos se describen a continuación, poniendo además en evidencia las ventajas y desventajas de su aplicabilidad:

1.1.1.1. Modelo Lineal Secuencial (Cascada)

Constituye uno de los primeros modelos utilizados en la construcción de software, del que partieron las demás metodologías; además es conocido como *el ciclo de vida clásico* por ser uno de los más antiguos modelos que aparecieron y que sigue cada uno de los pasos del ciclo de vida del desarrollo del software.

El modelo Cascada, según lo define Sommerville (2004, pág. 61) “considera las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, tales como la especificación de los requerimientos, el diseño del software, la implementación, las pruebas, etc.”

Mientras que según Pressman (2005, pág. 50) “sugiere un enfoque sistemático, secuencial hacia el desarrollo del software, que se inicia con la especificación de requerimientos del cliente y que continúa con la planeación, el

modelado, la construcción y el despliegue para culminar en el soporte del software terminado”.

En consecuencia, el modelo Lineal Secuencial o Cascada es una de las más antiguas metodologías que sigue las fases del ciclo de vida de software, es decir que parte de la especificación de los requerimientos por parte del cliente, diseño del software, implementación del sistema, pruebas y mantenimiento del mismo.

Para ilustrar el modelo Cascada, la tabla a continuación señala ventajas y desventajas:

Tabla 1: Ventajas y desventajas del Modelo Cascada

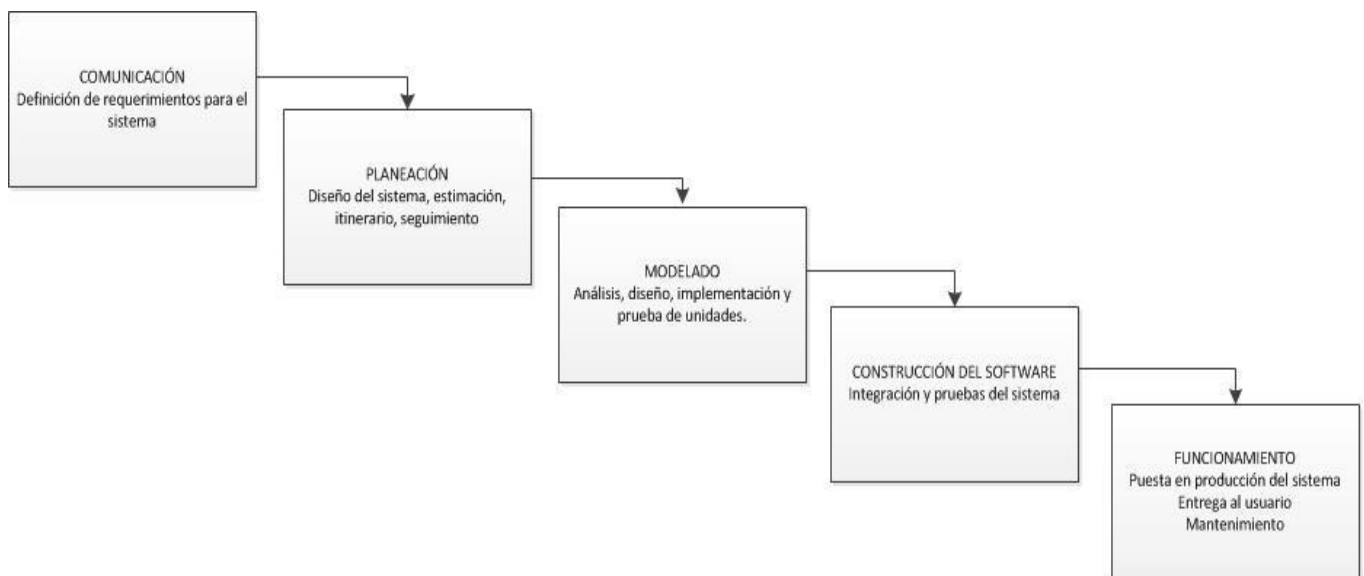
| VENTAJAS | DESVENTAJAS |
|---|---|
| Permite medir el tiempo de entrega del producto, evitando así el incremento del costo del producto por la demora incurrida | Dificulta dividir el proyecto en fases para su desarrollo, lo que impide realizar modificaciones conforme lo requiera el cliente |
| Permite el uso de tecnología conocida, por lo que se reduce el riesgo en el desarrollo | Demora en recorrer el ciclo de desarrollo del producto |
| Modelo de desarrollo sencillo, por cuanto utiliza las etapas o pasos básicos necesarios al momento de desarrollar el producto | Se efectúa el mantenimiento en el código fuente del programa |
| | Dificultad en realizar revisiones cuando los proyectos son extensos |
| | Los resultados sólo se los puede conseguir al final del proyecto. Errores no visualizados desde el comienzo, sería un desastre para el éxito del proyecto |

Elaborado por: la autora

Fuente: Cabrera et al. (2009)

Como toda metodología, el modelo Cascada tiene sus aspectos positivos y negativos al momento de utilizarla para el desarrollo de un proyecto. A pesar de ser de fácil utilización porque sigue el ciclo de vida básico de desarrollo del software, se corre el riesgo de que los resultados del proyecto no sean los óptimos ya que éstos sólo se los puede evidenciar al final del mismo. Y eso sin contar con posibles mejoras o cambios solicitados por el cliente y con que no existan errores desde el inicio del proyecto.

Gráfico 1: Modelo Cascada



Elaborado por: la autora

Fuente: Pressman (2005) /Sommerville (2004)

Como se puede inferir, el gráfico 1 especifica las fases de desarrollo del software, que son las mismas del modelo Cascada, y cuyo proceso se realiza de manera lineal y secuencial, esto es, cada paso del proceso sigue un orden y es consecuencia del paso anterior. Si el ciclo de vida del software empieza con

la especificación de los requerimientos (análisis), le sigue a esto la planeación del sistema (diseño), continuando con el modelado de los datos, el desarrollo y pruebas del sistema y, finalmente, la puesta en marcha del producto, este modelo de desarrollo permite trabajar sobre datos y metodología conocida para la entrega oportuna del producto desarrollado.

1.1.1.2. Modelo de Construcción de Prototipos

El modelo de Construcción de Prototipos sigue los siguientes pasos (Sommerville, 2004, pág. 62):

Identificación de requerimientos del sistema. Usuario y desarrollador trabajan en conjunto para conseguir determinar los requisitos del sistema a satisfacer. Se requiere definir los fines que persigue el sistema, y el alcance de su capacidad.

Desarrollo de un modelo que funcione. Se debe explicar a los usuarios qué método se va a utilizar, actividades que se van a desarrollar, qué secuencia se va a llevar a cabo, cuál será la responsabilidad de cada uno de los participantes. Es necesaria la formulación de un plan general para conocer cuál es el proceso del desarrollo del proyecto.

Es necesario establecer un calendario de trabajo para el comienzo y término de la primera iteración, en donde se describan los reportes y documentos que el sistema entregará y sus formatos. Además, se deben evaluar los costos que conlleva el desarrollo del prototipo.

Para el desarrollo del prototipo es necesario lo siguiente:

- Determinar el lenguaje de diálogo entre usuario y sistema
- Diseñar las pantallas y formatos que se utilizarán para la entrada de los datos
- Determinar los módulos fundamentales de procesamiento

- Especificar la salida del sistema
- En la fase de desarrollo del prototipo no se elabora documentación ni especificaciones de salida o de diseño de software.

Utilización del prototipo. El usuario es el encargado de probar el prototipo y evalúa sus características. Esto permite establecer si se darán cambios o modificaciones, o descartar características no necesarias en el sistema.

Revisión del prototipo. Se evalúa el prototipo y con los datos obtenidos se especifican las características que deberá tener la siguiente versión del prototipo.

Durante la evaluación se sondean las cualidades o peculiaridades de quienes van a ser los usuarios y la organización que poseen el dominio de la aplicación y la implementación de la misma.

Las modificaciones en el prototipo necesitan de planificación con el usuario antes de que sean entregados al analista.

¿Está el prototipo terminado? Se realizan 4 ó 6 iteraciones de los pasos anteriores una vez que tanto usuario y desarrolladores han concordado que el sistema ha madurado lo suficiente con todas las características adecuadas.

Una vez terminado el prototipo, se requiere decidir cómo se va a proceder. Para esto, hay cuatro opciones (Aparicio, 2012, págs. 22-24):

- Se abandona la aplicación, descartando el prototipo y la aplicación. El prototipo estableció que la aplicación o el enfoque utilizado para el proyecto son inadecuados y no justifica que se realice un nuevo desarrollo

- Se implanta el prototipo, por cuanto sus características cubren todos los requerimientos del usuario tanto permanente o como para futuro
- Se vuelve a desarrollar la aplicación, ya que el prototipo suministró información para definir las características básicas de la aplicación. Esta información sirve de base para el desarrollo de la aplicación con la finalidad de que ésta haga mejor uso de los recursos disponibles
- Se comienza un nuevo prototipo, ya que los datos suministrados por el primer prototipo propone otros datos y opciones. A partir de esta información se comienza con la construcción de un nuevo prototipo, diferente al anterior al que se le adicionan nuevas características y requerimientos de la aplicación

Por su parte Cabrera et al. (2009, pág. 4) señala que el prototipo forma el resultado último y “permite efectuar un test sobre determinados atributos del mismo sin necesidad de que esté disponible (...) A medida que el proceso progresa y el producto se completa, el prototipo ha de abarcar, cada vez más las características del producto final”.

Al analizar a ambos autores, es clara la afirmación de que para la aplicación de esta metodología es necesario el desarrollo e implementación de un prototipo que tenga las características de la aplicación que el cliente necesita. Una vez que el prototipo entre en funcionamiento, el usuario estará en la capacidad de determinar si la aplicación cumplirá con todos los requerimientos solicitados.

La técnica del prototipo se repite algunas veces. Esto permite al usuario y al desarrollador filtrar errores y construir un producto acorde a las necesidades del usuario del sistema.

Algunas ventajas y desventajas del modelo de Construcción de Prototipos son los que se presentan en la tabla 2:

Tabla 2: Ventajas y desventajas del modelo de Construcción de Prototipos

| VENTAJAS | DESVENTAJAS |
|---|---|
| Útil para cuando se desarrolla un producto que requiere del uso de nueva tecnología | En casos experimentales no funciona |
| Permite la retroalimentación por parte del usuario final del sistema; asimismo, el desarrollo es más rápido | Es lenta la gestión del desarrollo, porque ésta se encuentra dando vueltas hasta que el usuario final ha quedado satisfecho con el producto o hasta que se le ha puesto un límite al desarrollo |
| El usuario final va conociendo poco el producto y se familiariza con él | Impide conocer con anticipación el tiempo del desarrollo de la aplicación |
| El usuario puede ver la funcionalidad de la aplicación, aun cuando no se haya terminado el desarrollo del prototipo | Es difícil y complejo de realizarlo |

Elaborado por: la autora

Fuente: Cabrera et al. (2009)

1.1.1.3. Modelo DRA (Desarrollo Rápido de Aplicaciones)

El modelo de Desarrollo Rápido de Aplicaciones es un modelo de proceso del software que destaca un ciclo de desarrollo corto. Este proceso posibilita al grupo desarrollador a crear un “sistema completamente funcional” que abarquen períodos cortos de tiempo (60-90 días).

Este modelo tiene algunas fases (Aparicio, 2012, págs. 25-27):

Modelado de gestión. Se responde a preguntas como: ¿Qué información lleva al proceso de gestión? ¿Qué tipo de información se genera? ¿Quién genera la información? ¿Quién recibe la información? ¿Quién la procesa?

Modelado de datos. Se definen las características de los objetos y sus relaciones.

Modelado del proceso. Los datos modelados se transforman para conseguir un flujo de información adecuado para implementar la función de gestión.

Generación de aplicaciones. Este proceso asume la utilización de técnicas de cuarta generación, para usar módulos o elementos de programas que ya existen o la creación de unidades o componentes que pueden ser reutilizables.

Pruebas y entrega. Mediante la reutilización se comprueban componentes de los programas, lo que permite reducir el tiempo de las pruebas.

De acuerdo a Pressman (2005, pág. 53), el modelo DRA “es una adaptación ‘a alta velocidad’ del modelo en cascada en el que se logra el desarrollo rápido mediante un enfoque de construcción basado en componentes (...) Cumple con actividades genéricas del marco de trabajo que ya se han presentado: la comunicación, la planeación, el modelado, la construcción, el despliegue”.

Por ende, se podría definir al modelo DRA como un modelo de procesos de software que se basa en un desarrollo ágil y en un período en tiempo corto, mediante el uso de las fases de modelado, tanto de objetos como de procesos. Este modelo permite un control por parte de desarrolladores y usuarios del sistema, de todos los cambios realizados durante el transcurso del desarrollo.

1.1.1.4. Modelos evolutivos del proceso del software

Los modelos de procesos evolutivos del proceso del software son iterativos y tienen como característica principal el desarrollo de versiones cada vez más completas del software, todas presentadas al usuario para recibir su retroalimentación y filtrando errores hasta que el modelo deseado está listo (Aparicio, 2012, pág. 27). Algunos de estos modelos son:

a) Modelo Incremental

Une el modelo Secuencial Lineal y el modelo de Construcción de Prototipos para permitir entregar un software en pequeñas partes conocidas como incrementos, cuyos resultados se los obtiene luego de realizar las sucesivas iteraciones. Para este modelo no es necesaria una secuencia lineal de pasos para el desarrollo; la aplicación se crea aumentando componentes funcionales al sistema de incrementos (Aparicio, 2012, pág. 28).

Este modelo se adapta a entornos de alta incertidumbre.

Los incrementos que se entregan constantemente son versiones no completas del sistema final, pero que sirven para obtener la retroalimentación del usuario y medir la funcionalidad del producto (Aparicio, 2012, pág. 28).

Tiene algunas ventajas y desventajas.

Tabla 3: Ventajas y desventajas del modelo Incremental

| VENTAJAS | DESVENTAJAS |
|---|--|
| Se entrega producto frecuentemente | Dificulta la evaluación del coste total de la de la aplicación |
| Hay mayor participación del usuario | Requiere experiencia por parte de los gestores del proyecto |
| El retorno de la inversión es mayor | La aplicación a sistemas transaccionales, pues éstos generalmente son integrados y operan como uno solo |
| Se disminuyen los riesgos | La planeación de este modelo requiere de una gran planificación |
| Los requerimientos pueden ser cambiados ya que, como se entrega producto con cierta frecuencia, en cada versión los requerimientos pueden variar | Como este modelo se fundamenta en la retroalimentación del usuario final, los avances en el desarrollo pueden tornarse demasiado lentos |
| Los costos disminuyen, ya que si el incremento que se ha presentado no cumple con lo solicitado por el cliente, aquél se lo desecha y se vuelve a la versión anterior | No es útil para desarrollos grandes, puesto que consume demasiados recursos y tiempo |
| El usuario puede trabajar en una versión del producto | Por necesitar la retroalimentación del usuario final, se adiciona un costo para la empresa, por cuanto los empleados dedicados a la evaluación del producto dejan de ser productivos |

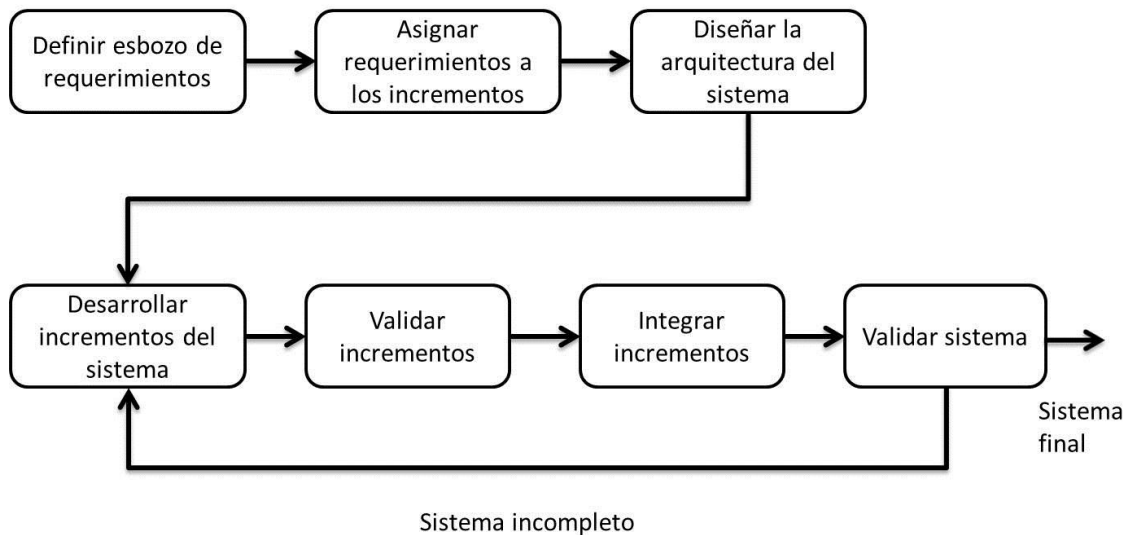
Elaborado por: la autora

Fuente: Cabrera et al. (2009)

Sommerville (2004, págs. 66-67) manifiesta que “en un proceso de desarrollo incremental, los clientes identifican, a grandes rasgos, los servicios

que proporcionará el sistema. (...) La asignación de servicios a los incrementos depende de la prioridad más alta entregados primero”.

Gráfico 2: Entrega incremental



Elaborado por: la autora

Fuente: Sommerville (2004)

Al realizar el análisis de los conceptos de modelo Incremental, se puede inferir que en este modelo de desarrollo la participación del usuario final es indispensable para la retroalimentación del sistema ya que, al ser el cliente parte activa en la evaluación del producto, se pueden realizar correcciones a los requerimientos mediante la utilización de los incrementos que son presentados por los gestores del proyecto para su revisión.

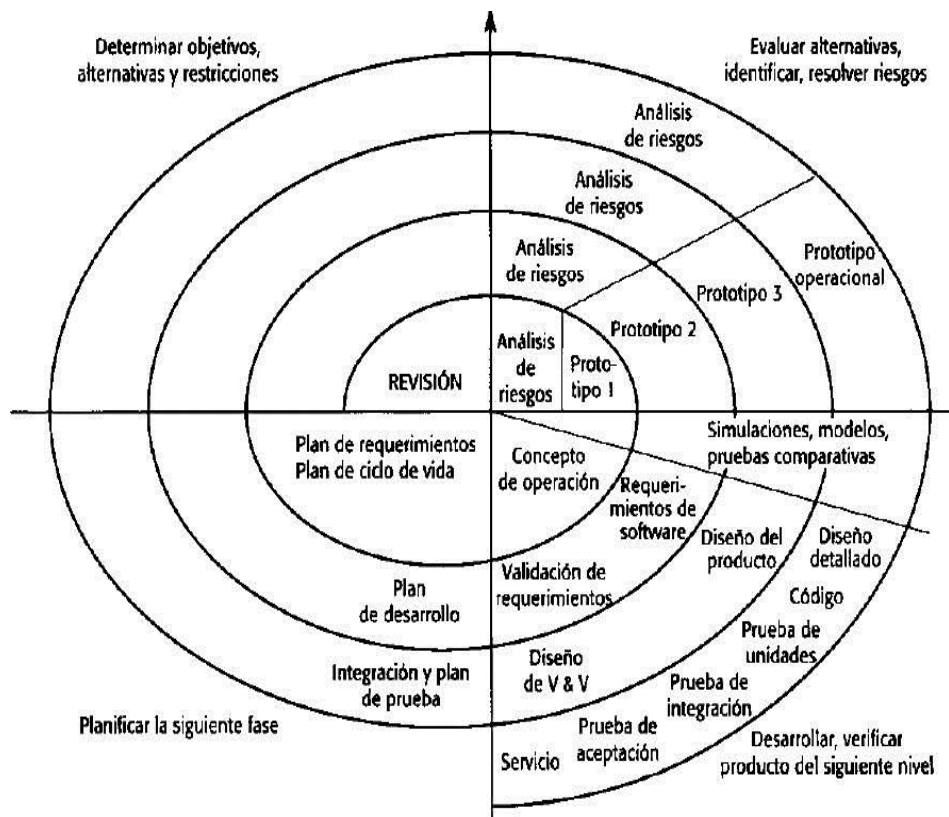
b) Modelo Espiral

Según Pressman (2005, pág. 58), el modelo en Espiral “es un modelo de proceso de software evolutivo que conjuga la naturaleza iterativa de la

construcción de prototipos con los aspectos controlados y sistemáticos del modelo en cascada. Proporciona el material para el desarrollo rápido de versiones incrementales del software”.

Por su parte Sommerville (2004, pág. 68-69) señala que el modelo espiral es una sucesión de “actividades con retrospectiva de una actividad a otra (...) empieza con la elaboración de objetivos, como el rendimiento y la funcionalidad. (...) se enumeran formas alternativas de alcanzar estos objetivos y las restricciones impuestas en cada una de ellas”.

Imagen 1: Modelo Espiral



Fuente: Sommerville (2004)

La imagen 1 muestra la secuencia seguida por el modelo Espiral. Ya se mencionó que este modelo de proceso de desarrollo sigue la secuencia del modelo Lineal o Cascada y, aunque la secuencia no retrocede al paso anterior, la representación del mismo se da como una espiral.

Los ciclos de la espiral tienen cuatro sectores (Sommerville, 2004, pág. 68):

Definir objetivos. Definición de objetivos específicos, identificación de riesgos del proyecto, planteamiento de estrategias alternativas.

Evaluar y reducir riesgos. Análisis pormenorizado de todos los riesgos que se han descubierto en el proyecto. Definición de estrategias para reducción de todos los riesgos existentes.

Desarrollar el sistema y validación del mismo. Elección de modelo de desarrollo del sistema.

Planificar. Revisión el proyecto y determinación de nuevos ciclos posteriores en la espiral, en caso de requerirse.

Por lo tanto, se define al modelo espiral como un modelo de proceso de desarrollo de software que se sustenta en otros modelos de procesos, uno de ellos como son la Construcción de Prototipos y el modelo Lineal o Cascada, ofreciendo al desarrollador las pautas necesarias para desarrollo incremental de versiones de software.

1.1.1.5. Metodologías ágiles

Este tipo de metodología pertenece al grupo de desarrollo ágil de software, que se fundamenta en la adaptación de eventuales cambios como forma de ampliar las posibilidades de éxito en la consecución de un proyecto de software (Marco Galindo & Marco Simó, 2010, pág. 194).

Los métodos ágiles dan prioridad a:

- Las personas y sus interacciones, que al desarrollo del sistema y las herramientas utilizadas
- Un software que sea funcional, que a una documentación
- El contacto y colaboración con el usuario, que a un contrato
- La respuesta a los cambios, que a una planificación (Marco Galindo et al., 2010, pág.194)

a) Programación extrema

Programación Extrema XP, según Canós, Letelier, & Penadés (2003, pág. 3) es un tipo de metodología ágil que se basa en “potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo”.

Esta metodología de desarrollo se fundamenta en la retroalimentación constante con el usuario y los desarrolladores, fácil comunicación entre todos los que forman parte del proyecto, soluciones sencillas para la implementación y audacia para hacer frente a los cambios (Canós et al., 2003, pág.3), a lo que Marco Galindo et al. (2010, 194-195) propone las siguientes técnicas:

Simple diseño. El diseño deberá ser lo más simple posible, en tanto se puedan resolver los problemas.

Realización de pruebas intensivas. Éstas son planificadas y redactadas antes del desarrollo para que el equipo de los programadores tenga en cuenta los errores en que se pudiera incurrir antes de empezar a trabajar en el código. Con esto se consigue que el código escrito sea verificado constantemente de modo que mejore en las pruebas previamente establecidas hasta después de realizados los cambios.

Refactoring. Búsqueda de mejora continua en el desarrollo del código. Hasta el código que ya se ha escrito y está aceptado como adecuado, puede ser mejorado, si hubiera la opción de hacerlo. Esto permite que se mejore la calidad de todo el sistema.

Programación en parejas. Se puede obtener calidad de código superior, si se trabaja en parejas: uno escribe el código, el otro lo revisa y viceversa.

La XP además sostiene que las variables que definen un proyecto como son coste en inversión y recursos, tiempo, calidad y conjunto de funcionalidades, únicamente tres pueden ser escogidas por el usuario y/o el jefe de proyectos. La otra variable es obligación exclusiva del grupo de desarrollo.

En definitiva, la metodología conocida como Programación Extrema o XP es una metodología de desarrollo de software; se enmarca dentro de las metodologías ágiles, en donde se hace énfasis en el desarrollo antes que en tratar de prever todo mediante documentación extensa. Toma las mejores prácticas de diferentes metodologías y las lleva al extremo con la finalidad de entregar un producto de calidad. Como todas las metodologías ágiles, es fácil adaptar cambios durante la implementación, porque a diferencia de otras metodologías no trata de proveer todos los requerimientos al inicio del desarrollo.

b) SCRUM

Según Díaz Polo & Delgado Dapena (2011, pág.11-12), Scrum es un “proceso de software iterativo utilizado en entornos basados en el desarrollo ágil de software, en el que se aplican de manera regular un conjunto de mejores prácticas para trabajar en equipo y obtener el mejor resultado posible de un proyecto”. Scrum señala el ámbito adecuado para la gestión de proyectos y está indicado para proyectos con un rápido cambio de requisitos.

Con Scrum se manejan usualmente todas las mejores prácticas para trabajo en equipo, y poder conseguir el mejor resultado posible al culminar un proyecto. Dichas prácticas se brindan soporte unas a otras y la elección de una de ellas se origina en un estudio de cómo trabajar en equipos altamente productivos.

Una de las características de Scrum es la entrega parcial y regular del producto final, las mismas que se seleccionan de acuerdo al beneficio que brindan al usuario final del proyecto. Por tal razón, Scrum es una metodología apropiada para desarrollo de proyectos en entornos complejos, en los que se necesita obtener resultados inmediatos, los requisitos cambian constantemente o poco definidos y, la innovación, la competitividad, la flexibilidad y la productividad son fundamentales para el éxito del proyecto (IBM DeveloperWorks, 2010).

Como se puede inferir tanto XP y Scrum, que son parte de las metodologías de desarrollo ágiles, son modelos de procesos que utilizan las mejores prácticas en el desarrollo de software, a fin de proporcionar al usuario final un producto acorde a las necesidades de los mismos. Este proceso se adapta a entornos complejos de desarrollo, teniendo como característica principal el trabajo en equipo y la retroalimentación del cliente.

1.1.1.6. RUP

El Proceso Unificado de Rational RUP, de acuerdo a Sommerville (2004, pág. 76) es un claro ejemplo de proceso “que proviene del trabajo en el UML y el asociado Proceso Unificado de Desarrollo de Software (...) Reúne elementos de todos los modelos de procesos genéricos, iteraciones de apoyo e ilustra buenas prácticas en la especificación y el diseño”.

Este proceso se lo representa habitualmente desde tres perspectivas (Sommerville, 2004, págs. 76-77):

- **Perspectiva Dinámica:** se presentan las distintas fases del modelo de proceso en el tiempo
- **Perspectiva Estática:** presenta las actividades del proceso que se representan
- **Perspectiva Práctica:** que apunta a las buenas prácticas que pueden ser empleadas mientras dura el desarrollo del proceso

El RUP tiene cuatro fases (Sommerville, 2004, págs. 76-77):

Fase de inicio: Se determina el entorno del negocio, es decir lo que interactuaría en él: individuos y sistemas existentes, para establecer cuáles serán las relaciones que se presentarán durante el desarrollo. Todos los datos que se obtengan serán puestos en consideración para calificar el aporte que el sistema a desarrollar le proporciona al negocio. Esto permite descubrir si la información que se obtiene es importante o no como para continuar o anular el proyecto.

Fase de elaboración: Se persigue comprender el dominio del problema, para lo que se requiere de un entorno arquitectónico adecuado para el sistema, la puesta en marcha del proyecto y descubrir los posibles riesgos que pudiera tener el mismo. Finaliza esta fase con el modelo de requerimientos necesarios para el funcionamiento del sistema, mediante la especificación de los casos de uso de UML, la especificación del entorno arquitectónico y la planificación del desarrollo.

Fase de construcción: Consiste en diseñar el sistema, el desarrollo del código y la realización de las pruebas. Para llegar a esta fase deberá desarrollar e integrar todas las partes o módulos del sistema; al término de la misma, deberá estar operativo el sistema y la documentación para los usuarios.

Fase de transición: Se busca trasladar el sistema desde el entorno del desarrollador al entorno del usuario para su ejecución en tiempo real. Al término

de esta fase el sistema deberá estar operativo, y con la documentación respectiva.

Por su parte Díaz Polo et al. (2011, pág. 7) señala que el RUP “ es un marco de trabajo genérico que puede especializarse en una gran variedad de sistemas de software (...)”. Este modelo de procesos hace uso del lenguaje de modelado UML para preparar todo lo necesario para el desarrollo de un sistema.

El mismo autor (Díaz Polo et al., 2011, pág. 7) manifiesta que el RUP “provee un enfoque para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su objetivo es asegurar la producción de software de alta calidad (...) dentro de un tiempo y presupuesto previsible”.

El RUP, como se puede determinar de las definiciones de los autores antes mencionados, es una metodología de desarrollo de software que utiliza las mejores prácticas de desarrollo de los procesos genéricos de software que permita obtener un producto de alta calidad. Se enfoca en el conocimiento del modelo de negocio de la organización y sus integrantes, para determinar la viabilidad o no del desarrollo del sistema. Además, utiliza el lenguaje de modelado UML para presentar los distintos escenarios y los actores que intervendrán en el sistema a desarrollar.

1.1.2 Pruebas de software

Una prueba de software es un análisis al término de las especificaciones del diseño y de la codificación. Es una serie de evaluaciones que se las lleva a cabo para encontrar errores.

Para el diseño de las pruebas de software se requieren técnicas que permitan:

- Comprobar si la lógica interna de los componentes del software es la adecuada
- Comprobar si los dominios de entrada y salida del programa son correctos. Así se pueden revelar errores tanto en la funcionalidad, como en el comportamiento y el rendimiento del sistema (Aparicio, 2012, pág. 110)

Es importante considerar objetivos, principios y características de las pruebas de software.

Son objetivos de las pruebas (Aparicio, 2012, pág. 110):

- Identificar los errores
- Presentar errores no identificados aún
- Identificar errores no detectados aún

Son principios de las pruebas de software (Aparicio, 2012, pág. 110):

- Seguimiento del software hasta llegar a los requerimientos del cliente
- Planificación previa de las pruebas antes de que se lleven a cabo
- Las pruebas absolutas del sistema no se las puede realizar

Son características de las pruebas (Pressman, 2005, págs. 419-420):

Facilidad de prueba. Permite descubrir si es fácil o no probar un programa.

Operatividad. Si el sistema a desarrollar se lo diseña e implementa con calidad, los posibles errores que impidan la realización de las pruebas serán escasos.

Observabilidad. Las entradas que son utilizadas como una parte de la prueba, tienen diferentes salidas. Tanto estados como variables del sistema desarrollado son evidentes y se los puede examinar mientras se ejecutan. Por

ende, una salida equivocada es fácil de identificarla. Errores internos del sistema son de fácil detección y reportados automáticamente. Hay accesibilidad al código fuente.

Controlabilidad. Control directo de estados y variables de software y hardware por parte del ingeniero que realiza las pruebas. Éstas pueden ser especificadas, automatizadas y reproducidas a conveniencia.

Capacidad para descomponer. Una vez que se ha logrado reconocer cuál es el alcance de la prueba a aplicar, se conseguirá aislar los problemas de forma mucho más rápida. Las pruebas volverán a ser aplicadas, pero ahora con más conocimiento de lo que se quiere conseguir con el sistema. Un software se construye de módulos independientes a los que se aplica la fase de pruebas de manera independiente.

Simplicidad. El sistema debe tener *simplicidad funcional* (características mínimas que satisfagan las necesidades del cliente), *simplicidad estructural* (arquitectura se presenta en módulos para restringir la expansión de fallas) y, *simplicidad de código* (adaptación a estándares de codificación para una fácil inspección y mantenimiento).

Estabilidad. Modificaciones en el programa no son comunes, las mismas se controlan cuando son evidentes y no interfieren en las pruebas existentes. El programa se recupera adecuadamente de las fallas.

Facilidad de comprensión. Tanto el diseño de la arquitectura del sistema como las dependencias entre componentes internos, externos y compartidos se perciben. Los manuales técnicos están perfectamente organizados, con información precisa y se encuentran al alcance del usuario. En caso de efectuarse modificaciones en el diseño del sistema, éstos son comunicados a los encargados de ejecutar las pruebas.

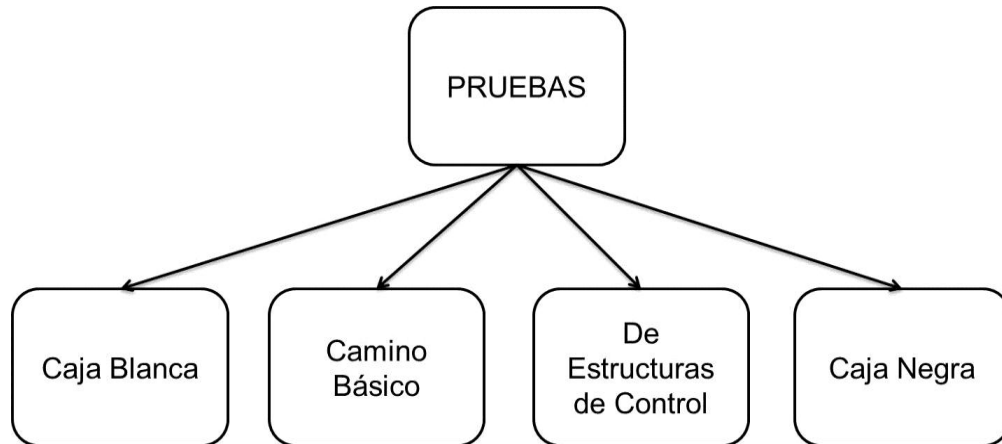
Se conoce, entonces, como prueba de software al análisis que se realiza al programa, el mismo que se lo realiza al final del desarrollo para tratar de descubrir posibles errores no descubiertos que se pueden presentar tanto en el diseño del sistema como en la codificación del mismo. Este análisis busca la comprobación de la lógica de los componentes de software, además de verificar la funcionalidad del sistema a implementarse.

Para la aprobación de un producto de software, es necesario diseñar previamente *casos de prueba*, que tienen como finalidad encontrar la mayor cantidad de fallas posibles con el menor esfuerzo y en el menor tiempo (Aparicio, 2012, pág. 112).

Por consiguiente, para aprobar un producto de software, según lo señala Aparicio (2012, pág. 112), se lo puede realizar de dos formas:

- Si se conoce la función del producto: se realizan pruebas para verificar su función operativa y tratar de encontrar errores en cada una de las funciones
- Si se conoce el funcionamiento del producto: se realizan pruebas que permitan verificar que todas las operaciones se encuentran conforme lo determinan las especificaciones; además, para verificar que todos los componentes internos del sistema tienen un correcto funcionamiento

Gráfico 3: Casos de Prueba



Elaborado por: la autora

Fuente: Aparicio (2012)

El gráfico 3 muestra los distintos tipos de pruebas que se practican para la verificación de los errores de software, que son tratados en los párrafos siguientes.

1.1.2.1 Prueba de Caja Blanca

Según Aparicio (2012, pág. 113), la prueba de la caja blanca “se centra en la estructura interna del programa. En este caso la prueba consiste en probar todos los posibles caminos de ejecución a través de las instrucciones del código que puedan trazarse”.

La autora mencionada señala además que, a través de esta prueba el ingeniero de software (Aparicio, 2012, pág. 113):

- Asegura que todas las rutas independientes que tiene cada uno de los módulos del sistema se los recorre, al menos, una vez
- Pasa por todas las decisiones lógicas de verdadero y falso
- Pasa por todos los bucles con sus respectivos límites operacionales

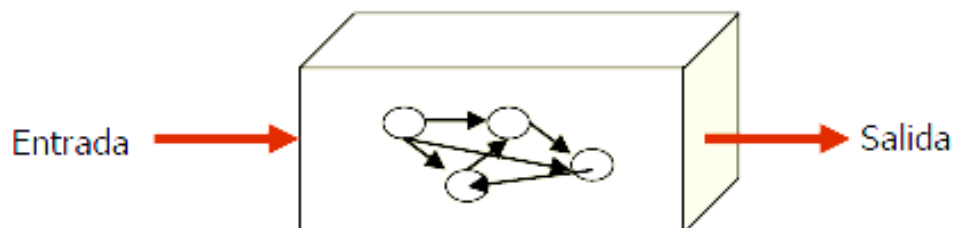
- Va por todas las estructuras internas de datos, lo que garantiza la validez de las mismas

Por su parte, Pressman (2005, pág. 422) manifiesta que la prueba de caja blanca para el software “se basa en un examen al detalle procedimental. Se prueban las rutas lógicas del software y la colaboración entre componentes, al proporcionar casos de prueba que ejerciten conjuntos específicos de condiciones, bucles o ambos”.

Además, a través de la prueba de caja blanca, el desarrollador puede derivar en casos de prueba que permitan asegurar que los caminos independientes dentro del módulo sean recorridos por lo menos una vez durante el proceso, ejecutándose también las decisiones lógicas de verdadero y falso junto a los bucles y las estructuras de datos internas que permitan garantizar la validez del proceso, según lo dice Pressman (2005, pág. 423).

Como ambos autores concuerdan en su definición de prueba de caja blanca, se puede definir a esta prueba como aquella que se aplica al proceso interno del software, para la verificación de las estructuras de datos, las estructuras lógicas del sistema, recorriendo por lo menos una vez las rutas de cada uno los módulos independientes para garantizar la completa validez de todo el proceso.

Imagen 2: Prueba de Caja Blanca



Fuente: Aparicio (2012)

La imagen 2 presenta el proceso de la prueba de Caja Blanca, en donde en la entrada ingresan los requerimientos para el sistema. En la parte propiamente del proceso, se realiza el recorrido de cada uno de los módulos independientes del sistema para su verificación, antes de que se produzca el resultado final, que será el sistema como tal.

1.1.2.2 Prueba de Caja Negra

La prueba de Caja Negra, de acuerdo a lo que dice Aparicio (2012, pág. 116), busca analizar “la especificación de las funciones, la entrada y la salida para derivar los casos. Aquí, la prueba ideal del software consiste en probar todas las posibles entradas y salidas del programa” (Aparicio, 2012, pág. 116).

La autora también señala que esta prueba busca errores en relación con (Aparicio, 2012, pág. 116):

- Funciones incorrectas o ausentes
- Fallas de interfaz
- Errores en las estructuras de datos o en los accesos a bases de datos externas
- Errores de rendimiento
- Errores de inicialización y de terminación (Aparicio, 2012, pág. 116)

Por su parte Pressman (2005, pág. 433-434) señala que la prueba de caja negra o “*pruebas de comportamiento*, se concentran en los requisitos funcionales del software. Permiten (...) derivar conjuntos de condiciones de entrada que ejercitarán por completo todos los requisitos funcionales de un programa”. El mismo autor señala que la prueba de caja negra no reemplaza a las pruebas de caja blanca, sino que presentaría un enfoque distinto a dichas pruebas, permitiendo que se puedan descubrir errores desde otra perspectiva.

El mencionado autor (Pressman, 2005, pág. 434) señala también que, mientras las pruebas de caja blanca se las lleva a cabo cuando se da inicio al proceso de pruebas, las de caja negra son aplicadas en las etapas finales del proceso de pruebas. Estas prueban responden las siguientes preguntas:

- ¿Cómo se logra probar la validez funcional del sistema?
- ¿Cómo se logra probar el comportamiento y el desempeño del sistema?
- ¿Cuáles son las clases de entrada que serán elegidas como buenos casos de prueba?
- ¿Es particularmente sensible el sistema a determinados valores de entrada?
- ¿Cómo se pueden aislar los límites de una clase de datos?
- ¿Qué tasas de datos y qué volumen tolera el sistema?
- ¿Cuáles son los efectos que tienen combinaciones determinadas de datos sobre la operación del sistema?

Se puede definir entonces como pruebas de Caja Negra a aquellas que permiten comprobar la validez del sistema mediante su aplicación al inicio y al final del proceso. Al inicio, para comprobar todas las entradas posibles del sistema y determinar si los requerimientos funcionales del mismo presentes en las entradas son los necesarios para el adecuado comportamiento del sistema; al final, para verificar cuáles son las salidas y definir, asimismo, si existen errores en el resultado de las funciones aplicadas a las distintas estructuras de datos existentes.

Imagen 3: Prueba de Caja Negra



Fuente: Aparicio (2012)

La imagen 3 muestra el proceso de la prueba de caja negra, en donde se realizan las verificaciones de las funciones internas para identificar los posibles errores en la salida del sistema.

1.1.2.3 Prueba de la Ruta Básica

De acuerdo a la definición propuesta por Pressman (2005, pág. 423), la prueba de la ruta básica “permita que el diseñador de casos de prueba obtenga una medida de complejidad de un diseño procedimental y que use esta medida como guía para definir un conjunto básico de rutas de ejecución”. Los casos de prueba registrados para ejercitar el conjunto básico deberán asegurar que las instrucciones del programa se ejecuten, por lo menos, una vez durante el desarrollo de las pruebas (Pressman, 2005, pág. 423).

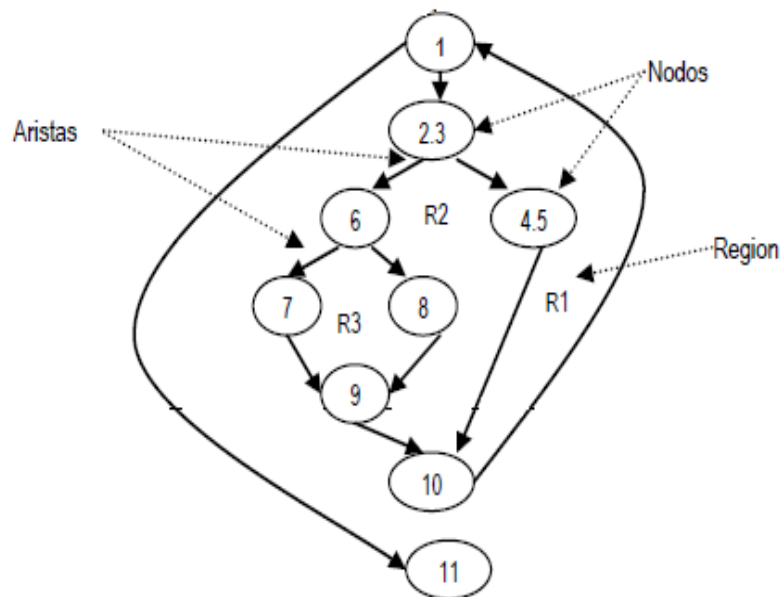
Antes de utilizar la prueba de la ruta básica se deben aplicar métricas para realizar mediciones que permitan comprender la complejidad de un programa.

Aparicio (2012, pág. 113-114) define a esta métrica como “una métrica que proporciona una medición cuantitativa de la complejidad lógica de un programa. (...) está basada en la teoría de grafos”.

El valor calculado mediante esta métrica determina el número de caminos independientes con los que se ejecuta un programa, definiendo un tope máximo superior que permita realizar las pruebas necesarias para avalar que las rutinas se hayan realizado por lo menos una vez (Pressman, 2005, pág. 426).

Se define, entonces, como prueba de la ruta básica a aquella que permite determinar los casos de prueba que aseguren la ejecución de las rutinas por lo menos una vez. Esta prueba delimita con exactitud lo complicado del procedimiento de diseño para que sirva de modelo y delinear rutas de ejecución.

Imagen 4: Prueba de la Ruta Básica



Fuente: Aparicio (2012)

La imagen 4 muestra los distintos nodos y los caminos que corresponden a las rutinas de ejecución.

1.1.2.4 Prueba de la Estructura de Control

Pressman (2005, pág. 430) señala que las pruebas de estructura de control tiene algunas variantes. Entre ellas se encuentra la prueba de la ruta básica que, no obstante de ser sencilla y práctica, por sí misma no es conveniente. Engloba tres pruebas (Aparicio, 2012, págs. 114-115-116):

Prueba de condición: Analiza todas las condiciones del programa para descubrir fallas en ellas y también fallas del programa.

Prueba del flujo de datos: Se concentra en la elección de rutas de prueba de un programa, conforme al sitio en donde se encuentren las definiciones y los usos de las distintas variables del programa. Es muy útil para elegir las rutas de pruebas de un programa que incluyan sentencias if o bucles anidados (Aparicio, 2012, págs. 114-115-116):

Prueba de bucles: Trata de comprobar la validez de las construcciones de bucles (Pressman, 2005, pág. 433):

a) Bucles simples, en donde:

- Se ignora por completo el bucle
- Se ignora una sola vez el bucle
- Se recorre el bucle en dos ocasiones
- Se realiza m pasos por el bucle con $m < n$
- Se efectúa $n-1$, n y $n+1$ pasos durante el recorrido por el bucle

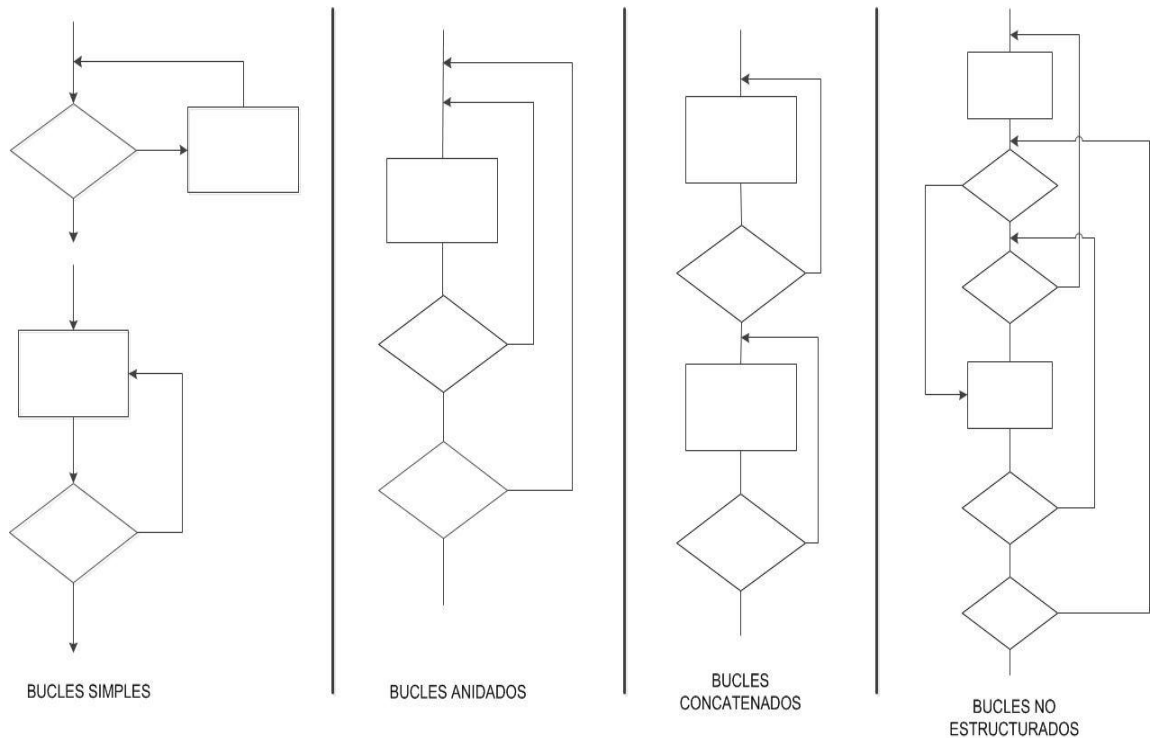
b) Bucles anidados, en donde se debe realizar:

- Empezar el recorrido por el bucle que se encuentra lo más interior posible
- Realizar pruebas de bucles simples para aplicarla al bucle que se encuentra lo más interior posible

- El progreso de las pruebas debe ser hacia afuera, llevándolas al siguiente bucle
 - Seguir con las pruebas hasta verificar todos los bucles
- c) Bucles concatenados**, en donde se prueban estos bucles con el método para bucles simples, que sean independientes. Si no son independientes, se hace uso del método para bucles anidados
- d) Bucles no estructurados**, los cuales deberán ser delineados nuevamente para descubrir “el uso de las construcciones de programación estructurada” (Pressman, 2005, pág. 433)

Como prueba de Estructuras de Control se entiende, entonces, a aquellas pruebas que se realizan a la estructura del programa. Tales pruebas son necesarias para el análisis de los errores que puedan tener las rutinas del programa, buscando al mismo tiempo las rutas de prueba del programa que incluyan sentencias if o también bucles.

Gráfico 4: Tipos de Bucles



Elaborado por: la autora

Fuente: Aparicio (2012) / Pressman (2005)

En el gráfico 4 se encuentran los distintos tipos de bucles y las rutinas asociadas a ellos para la aplicación de las pruebas de estructuras de control.

1.1.3 Métricas de software

Aparicio (2012, pág.42) señala que la “métrica del software comprende una gama de mediciones para el software que se aplican tanto al proceso como al proyecto de software para ayudar en la estimación, el control de calidad, evaluación de productividad y el control de proyectos” (Aparicio, 2012, pág. 42).

Existen motivos para realizar la medición de los procesos de software, los productos y los recursos (Aparicio, 2012, pág. 42):

- Caracterizar: Necesario para la comprensión de los procesos, productos, recursos y el entorno de desarrollo
- Evaluar: Necesario para establecer el estado de los procesos en relación con el diseño
- Predecir: Sirve para la planificación
- Mejorar: Para optimizar la eficacia del producto y la productividad de los procesos del software

Por su parte Pressman (2005, pág. 468) señala que se debe diferenciar medida y métrica. Una medida “proporciona un indicador cuantitativo de la extensión, la cantidad, la dimensión, la capacidad o el tamaño de algún atributo de un producto o proceso” (Pressman, 2005, pág. 468).

El autor antes mencionado (Pressman, 2005, pág. 468) señala como métrica una “medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado”. Por lo tanto, el experto en software reúne medidas y crea métricas, que le permitirán conseguir los indicadores. Como indicador el mismo autor (Pressman, 2005, pág. 468) define como “una métrica o una combinación de métricas que proporcionan conocimientos acerca del proceso del software, un proyecto de software o el propio producto. (...) proporciona conocimientos que permiten (...) ajustar el proceso, el proyecto o el producto para que las cosas mejoren” (Pressman, 2005, págs. 468-469).

En conclusión, se conoce como métricas de software a las distintas formas de medición del producto del software que son desarrolladas por los ingenieros de sistemas con la finalidad de conocer la calidad del producto que se ofrece y que brindan pautas para la mejora del mismo.

1.1.3.1 Métricas para el modelo de análisis

Este tipo de métricas, de acuerdo a Pressman (2005, pág. 474), son aquellas que buscan explorar el modelo del análisis del software para tratar de prever el tamaño que tendrá el software. Consideran aspectos como:

Métricas basadas en la función. Brinda una medida indirecta (punto de función) en relación con la funcionalidad que se empaqueta y entrega el software. Las métricas de puntos de función se utiliza para (Pressman, 2005, pág. 474):

- Evaluar costos o esfuerzo que se necesita en el diseño, la codificación y la aplicación de las pruebas de software
- Pronosticar la cantidad de posibles errores que aparecerán durante la aplicación de las pruebas
- Prever la cantidad de componentes, líneas de código planificadas, o ambas, en el sistema implementado

Los puntos de función son valores basados en medidas contables del dominio de la información del software y las evaluaciones de la complejidad de éste. Estos valores están definidos como (Pressman, 2005, págs. 474-475):

- *Número de entradas externas:* se producen de dos formas desde un usuario o transferida desde otra aplicación, ofreciendo datos destinados para la aplicación o información de control
- *Número de salidas externas:* se producen en el interior de la aplicación y brindan datos al usuario sobre informes, pantallas, mensajes de error, etc.
- *Número de consultas externas:* es una entrada en línea para generar respuesta del software
- *Número de archivos lógicos internos:* son grupos lógicos de datos que se encuentran dentro de los límites de las aplicaciones

- *Número de archivos de interfaz externos:* grupos lógicos de datos que se encuentran fuera de la aplicación y que ofrecen nuevos datos que se pueden utilizar dentro de la misma (Pressman, 2005, págs. 474-475)

Tamaño del sistema: se refiere a la medición del tamaño general del sistema, que se lo define desde el punto de vista de la información que se encuentra disponible, como parte del modelo de análisis (Pressman, 2005, pág. 472)

Métricas para calidad de la especificación: indica el grado con que se han especificado los requisitos del sistema, como son:

- Especificidad (falta de ambigüedad)
- Estado del avance del desarrollo
- Corrección en las especificaciones
- Comprensión de requisitos
- Verificación de requisitos
- Consistencia interna y externa
- Facilidad para alcanzar los objetivos
- Requerimientos concisos
- Facilidad para seguimiento del software
- Facilidad para posibles cambios y modificaciones
- Precisión en los requisitos
- Facilidad de reutilización del software resultante (Pressman, 2005, págs. 477-478)

De lo anotado en líneas anteriores, las métricas para el modelo de análisis del software son aquellas que le permiten al grupo de proyectos estimar cuál será el tamaño del software a desarrollar, considerando que para poder determinar cuál será la extensión del programa se deben tomar en cuenta

aspectos relacionados con: costos, posibles errores, líneas de código a desarrollar, entradas y salidas del sistema, consultas, para proveer datos útiles al mismo y permitir, al mismo tiempo, conocer cómo se han recibido los requisitos para el desarrollo de la aplicación.

1.1.3.2 Métricas para el modelo de diseño

Como manifiesta Pressman (2005, pág. 472) este tipo de métricas cuantifican los atributos del diseño de tal forma que el desarrollador o líder del proyecto puede evaluar el diseño del mismo. Esto incluye:

Métricas del diseño arquitectónico: Se muestran vestigios sobre la calidad del diseño arquitectónico y lo efectivo que pueda resultar los módulos o componentes dentro del sistema. Estas son pruebas de caja negra (Pressman, 2005, pág. 479).

Métricas de diseño al nivel de componente: Evalúan la complejidad de los componentes de software y otros aspectos que son de impacto para la medición de la calidad. Se incluyen medidas de cohesión, acoplamiento y complejidad del módulo (Pressman, 2005, pág. 488).

Para esto, es necesario conocer cómo funciona internamente el módulo que se está considerando para la aplicación de las métricas y se las aplica cuando ya se encuentra desarrollado el diseño procedimental.

Métricas orientadas a la operación: Este tipo de métricas miden el contenido de la estructura de conectividad antes que los módulos individuales del sistema. “Se apreciarán mejor las cosas si se examinan las consultas promedio de métodos (operaciones)” (Pressman, 2005, pág. 491). Tres métricas son consideradas importantes en este grupo (Pressman, 2005, pág. 491):

- *Tamaño promedio de operación*: número de mensajes enviados por la operación permite advertir el tamaño de la operación
- *Complejidad de la operación*: se aplica cualquier métrica de complejidad adaptada al software
- *Número promedio de parámetros de la operación*: cuantos más parámetros existan en la operación, será más compleja la colaboración entre los objetos. Por este motivo es necesario que el número promedio de parámetros de la operación estén en un nivel lo más bajo posible

Métricas de diseño de la interfaz: (Pressman, 2005, pág. 492)

Relacionadas con la facilidad de uso del sistema

Métricas especializadas en diseño orientado a objetos: Se encarga de medir las particularidades de las clases, así como lo relacionado con la comunicación y la colaboración. Se deben tomar en cuenta algunas características en cuanto a las métricas del diseño orientado a objetos: tamaño, complejidad, acoplamiento, suficiencia, grado de avance, cohesión, primitivismo, similitud, volatilidad (Pressman, 2005, págs. 481-482-483).

1.1.3.3 Métricas para el código fuente

Pressman (2005, pág. 472) señala que “estas métricas miden el código fuente y se usan para evaluar su complejidad (...)”. También permite probar características como (Pressman, 2005, pág. 472):

- Métricas de complejidad: se refieren a la medición de la complejidad lógica del código fuente
- Métricas de longitud: brindan una idea del tamaño del software

1.1.3.4 Métricas para las pruebas

Este tipo de métricas, según Pressman (2005, pág. 473), facilitan al desarrollador el diseño de casos de prueba seguros y ayuda a evaluar la eficacia de las pruebas que se apliquen al software desarrollado. Son (Pressman, 2005, pág. 473):

- Métricas de cobertura de instrucciones y ramas: conducen al diseño de casos de prueba que proveen protección al programa
- Métricas relacionadas con los defectos: su aplicación va más a la búsqueda de defectos que a las pruebas en sí
- Efectividad de la prueba: indican, en tiempo real, la efectividad de las pruebas que se aplican
- Métricas en el proceso: se relacionan con el proceso y se las determina conforme se vayan aplicando las pruebas

1.1.4 Gestión de proyectos

Grompone (1996, pág. 10) señala que la gestión de proyectos de software tiene como finalidad los mismos objetivos que puedan tener todas las gestiones de proyectos en el área de la ingeniería:

- Evaluar lo que sucederá con el desarrollo de un proyecto nuevo
- Evaluar lo sucedido con un proyecto ya finalizado

Ambas apreciaciones deberán responder preguntas cuantitativas como (Grompone, 1996, pág. 10):

- ¿Cuál será el plazo de entrega del proyecto?
- ¿Qué cantidad de personal se requiere para el desarrollo del proyecto?
- ¿Cuál será el costo del proyecto?

Por su parte Aparicio (2012, pág. 35) considera que la gestión de proyectos “implica la planificación, supervisión y control de personal, procesos y eventos, mientras que evoluciona el software”.

La autora señala que la gestión de proyectos básicamente se concentra en 4P's:

- *Personal*, referente con la necesidad de individuos que sean parte del grupo de desarrolladores del software
- *Producto*, relacionado con los objetivos y entorno del software
- *Proceso*, en relación con la estructura del plan para el desarrollo del software del sistema
- *Proyecto*, en relación con la planificación y control de los proyectos de software

1.1.4.1 El personal

Sommerville (2004, pág. 544) manifiesta que el personal de una empresa de software “representa el capital intelectual, y es misión de los gestores asegurar que la organización obtenga los mejores beneficios posibles al invertir en las personas”. Esto se cumple en las empresas en donde el éxito es visible y los integrantes de la misma son respetados dentro de la organización, a las cuales se deberá reconocer los logros obtenidos mediante premios de acuerdo a sus capacidades (Sommerville, 2004, pág. 544)

Por su parte Aparicio (2012, pág. 36) dice que el personal es el “recurso humano que participa y colabora en el proceso del software y su organización para el desarrollo de proyectos de software de manera eficaz” (Aparicio, 2012, pág. 36).

Dicha fuente señala además que el personal está dividido en algunas categorías (Aparicio, 2012, pág. 36):

Participantes. A su vez, tiene una clasificación:

- *Gestores superiores*, que son los que tienen a su cargo la definición de los aspectos coyunturales del negocio
- *Gestores técnicos del proyecto*, que tienen a su cargo programar, impulsar, estructurar y examinar a los técnicos que van a participar en el proyecto de desarrollo de software
- *Profesionales*, que tienen la obligación de proveer todas las capacidades técnicas para la ingeniería del sistema o aplicación
- *Clientes*, que son los usuarios del sistema, los que deberán proporcionar los requerimientos necesarios con que funcionará el sistema
- *Usuarios finales*, que son los que tienen a su cargo la interacción con el sistema

Jefe de equipo. Es el encargado de la gestión de los proyectos de software, el mismo que:

- Está encargado de evaluar todos los aspectos técnicos y de organización más importantes para el desarrollo del sistema
- Está en la capacidad de responsabilizarse del control del proyecto y permitir sugerencia de ideas de los elementos más notables del grupo de proyecto
- Da impulso y motiva las iniciativas y resultados de los integrantes del grupo de trabajo
- Especifica que la calidad del producto a desarrollar es importante

Equipo de software. Tres niveles para la organización del equipo de trabajo:

- *Descentralización democrática.* El equipo a constituirse no tiene un jefe estable, se eligen coordinadores del equipo a corto plazo. Las resoluciones que se toman en relación a un determinado tema en particular es tomada en consenso con todos los miembros del equipo. El tipo de comunicación en el grupo es horizontal
- *Descentralización controlada.* El grupo tiene un líder o jefe inmediato que es el encargado de la organización de un trabajo en particular, y jefes menores para la organización de trabajos menores. Las decisiones sobre resolución de problemas las hace el grupo, pero la implementación de las mismas se las realiza por subgrupos liderados por su jefe. La comunicación entre personas y subgrupos es horizontal, aunque existe comunicación vertical durante toda la jerarquía de control
- *Centralización controlada.* La solución de problemas jerárquicos está en manos del jefe o líder del equipo, así como también la organización interna del grupo de trabajo. Aquí, la comunicación entre jefe e integrantes del grupo de trabajo es vertical

Coordinación y comunicación. Se busca medios de comunicación para la coordinación del grupo de trabajo. Es necesario:

- *Comunicación formal.* Se realiza de manera escrita, mediante asambleas y otros medios de comunicación. La documentación contiene archivos de ingeniería de software, notas técnicas, informes de seguimiento del trabajo
- *Comunicación informal.* Se refiere a una comunicación más personalizada. Contiene reuniones de trabajo del grupo para el conocimiento de nueva información que permita resolver problemas pendientes

- *Comunicación electrónica.* Correos electrónicos, boletines, audio conferencias, videoconferencias (Aparicio, 2012, págs. 36-37)

Como se puede inferir en los datos obtenidos de ambos autores, la selección del personal para la conformación de un equipo de trabajo que tenga a su cargo la realización de un proyecto, es el recurso humano que debe ser cuidadosamente escogido para la puesta en marcha de dicho proyecto. Escoger un equipo que sea eficiente en su trabajo implica conocer a fondo al personal del que se dispone, para saber asignar tareas y responsabilidades a líderes de equipo como para elegir el grupo de desarrollo como tal que va a trabajar en el proyecto, y también para determinar cuáles serán los canales de comunicación más adecuados que permitan la coordinación del trabajo en equipo.

1.1.4.2 El producto

Cuando se piensa emprender un proyecto, lo primero que debe determinar son los objetivos y el entorno que va a servir como marco de referencia para el producto final. A más de esto, es necesario estimar soluciones alternas y definir todo tipo de restricción técnica y de gestión. De no tener estos reportes no se puede determinar costos, riesgos, asignación real de trabajos o un calendario cómodo que refleje los avances del proyecto que está en curso (Pressman, 2005, pág. 642).

Es importante que el gestor del proyecto y el futuro usuario del sistema tengan una reunión previa para determinar objetivos y entorno del producto. Los objetivos definen qué es lo que busca conseguir la empresa con el producto final vistos desde la óptica del cliente sin considerar la forma en que se logrará conseguirlos. El entorno determina los datos primarios, funciones y comportamientos propios del producto y, aún más importante todavía, los intentos por conectar las características del producto de forma cuantitativa.

Cuando los objetivos y el entorno ya están claros se precisa plantear soluciones alternativas (Pressman, 2005, pág. 642).

Aparicio (2012, pág.38) sugiere que, en cuanto al producto, hay que conocer dos aspectos: el ámbito y la descomposición del problema. El **ámbito** determina:

- *Contexto*: sobre cómo se acopla el programa que se va a desarrollar en el sistema o ambiente empresarial y cuáles son las limitaciones propias de ese ambiente
- *Objetivos de información*: sobre cuáles son los objetos de datos de ese software que son evidentes al usuario o cuáles son los objetos necesarios para la entrada
- *Función y rendimiento*: sobre cuáles son las funciones que realiza ese programa como medio para modificar los datos de entrada en una salida y cuáles son las características de rendimiento especiales planteadas

La **descomposición del problema** se refiere al estudio de los requerimientos del programa y se aplica en:

- La funcionalidad del programa que se va a desarrollar y que se debe entregar al usuario final
- Los procesos necesarios para la entrega (Aparicio, 2012, pág. 38)

Se concluye de los mencionados autores que el producto es el resultado final del programa desarrollado por el equipo de trabajo, una vez analizados los requerimientos y las necesidades del usuario que plantea el problema. A este producto se le deberá determinar el entorno en donde va a ser utilizado junto con los objetivos y requerimientos de entrada y, descomponer el problema para conocer la funcionalidad del mismo y los procesos necesarios para la solución del problema.

1.1.4.3 El proceso

Ya en materia del proceso como tal, Aparicio (2012, pág.39) señala que el encargado del proyecto determina “qué modelo de proceso es el más adecuado para:

- Los clientes que han solicitado el producto y la gente que realizará el trabajo
- Las características del producto
- El entorno del proyecto” (Aparicio, 2012, pág. 39)

Además, en el proceso se debe considerar (Aparicio, 2012, pág. 39):

1) Maduración del problema y el proceso, en donde una vez que todo el grupo de trabajo que va a desarrollar el software, tiene como tarea definir una serie de actividades con las cuales poder trabajar en cada función del problema entregado. Las actividades consideran:

- *Comunicación*: Se determina el tipo de comunicación que se debe establecer entre el desarrollador y el usuario final, para conseguir los requerimientos del sistema
- *Planificación*: Está constituida por el grupo de tareas cuya finalidad es determinar cuáles son los recursos y la planificación temporal del proyecto
- *Análisis del riesgo*: Está definido por tareas que se necesitan para determinar y valorar los riesgos técnicos del proyecto y la gestión del mismo
- *Ingeniería*: Está definida por las tareas necesarias para construir uno o más prototipos de la aplicación

- *Construcción y entrega:* La constituye el grupo de tareas necesarias para construir, probar, instalar y brindar asistencia técnica al cliente
- *Evaluación del cliente:* La constituye el grupo de tareas encaminadas a que el cliente o usuario final valore las representaciones del programa creadas durante la fase de ingeniería

2) Descomposición del proceso, en donde las distintas actividades de comunicación, planificación, análisis de riesgo, ingeniería, construcción, entrega y evaluación se acoplan el modelo de desarrollo de software elegido por el equipo y el gestor de proyectos (Aparicio, 2012, pág. 39).

Pressman (2005, pág. 642) manifiesta que en lo relacionado al proceso, éste brinda el ámbito de trabajo referencial a partir del cual se puede determinar el plan detallado para la realización del desarrollo. Pocas de las actividades que se concentran en el ámbito de trabajo son aplicables a los proyectos de software en general, sin que tenga importancia el tamaño o la complejidad de dichos proyectos. Ciertas labores (tareas, hitos, productos de trabajo y puntos de control de calidad) son las que hacen que todas las actividades que se desarrollan en el marco de trabajo se adecúen tanto a las características del proyecto de software como a los requerimientos del equipo de trabajo. Por último, las actividades de protección del software (control de calidad, gestión de configuración y medición) resguardan el modelo del proceso. Las actividades de protección son autónomas de toda acción del marco de trabajo y sucede en el transcurso del proceso.

Por tanto, en la gestión de proyectos, un proceso de software le permite al gestor del proyecto definir cuál son las características, usuarios y campo de acción en que éste se va a desarrollar, para establecer las actividades idóneas

que se ajusten a las necesidades del mismo teniendo en cuenta cuán complejo sea el proyecto. Estas actividades deben ser conocidas por el grupo de desarrollo con la finalidad de establecer un marco de referencia para que tareas y características inherentes al proyecto se ajusten tanto a las necesidades y requisitos propios del equipo de trabajo como al modelo de desarrollo elegido. Existen características definidas por el grupo de trabajo como son las tareas de maduración del proyecto, esto es, las que son definidas al inicio del desarrollo y consideran actividades como comunicación, planificación, análisis del riesgo, ingeniería, construcción y entrega, evaluación del cliente. Pero también hay tareas de protección del software como son control de calidad, gestión de configuración y medición, que son características independientes del marco de referencia en que se desarrolla el proyecto, las mismas que se producen durante todo el proyecto.

1.1.4.4 El proyecto

Aparicio (2012, pág.40) señala que para tener éxito en un proyecto, es necesaria la gestión de proyectos de calidad. Esto se consigue cuando:

- Se conoce el problema en su totalidad y se determinan los objetivos que se quieren alcanzar
- Se establece un ambiente de trabajo amigable con el equipo de trabajo y se proporcionan incentivos
- Se controla la calidad del producto a través del seguimiento constante de todas las tareas y actividades que se llevan a cabo durante el desarrollo del proyecto
- Las decisiones en relación con las actividades y el desarrollo del proyecto son tomadas en consenso entre el gestor del proyecto y el equipo de trabajo
- Se realiza una evaluación de la planificación inicial para compararla con la planificación real de los procesos, a fin de que se analicen

métricas del proyecto y obtener información para la realimentación de dichos procesos

Para Pressman (2005, pág. 643) “los proyectos de software se realizan de manera controlada por una razón principal: es la única forma conocida de gestionar la complejidad”. El autor señala además que si se quiere eludir que el proyecto fracase, el gestor del proyecto y el grupo de ingenieros de software que son los que desarrollan el producto final “deben eludir un conjunto de señales de advertencia comunes, comprender factores de éxito críticos que conducen a una buena gestión del proyecto y desarrollar un enfoque de sentido común para planificar, supervisar controlar el proyecto” (Pressman, 2005, pág. 643).

Es importante señalar algunas señales que son indicadores de que un proyecto puede fracasar (Pressman, 2005, pág. 656):

- El equipo de desarrollo no comprende cuáles son los requerimientos del cliente
- El ámbito del producto no se encuentra definido claramente
- Hay mala gestión de cambios
- Se cambia a otra tecnología de desarrollo
- Hay cambio o mala definición de necesidades comerciales
- Tiempo de entrega del proyecto no realista
- Hay resistencia de los usuarios a colaborar con el equipo de trabajo
- El patrocinio que tiene el proyecto se pierde o no fue aplicado adecuadamente
- Hay mala elección de los miembros del equipo de desarrollo porque sus habilidades no son las adecuadas

- Tanto gestores de proyecto como técnicos de desarrollo eluden el uso de mejores prácticas y las lecciones que se aprenden durante el desarrollo del proyecto

Para contrarrestar estas señales de fracaso, se presenta el siguiente enfoque (Pressman, 2005, págs. 656-657):

- *Comenzar con pie derecho*: Es necesario trabajo muy duro para comprender el problema a resolver. De este modo se pueden determinar objetivos y perspectivas reales para los participantes en el proyecto, lo que se consigue con la conformación de un equipo de trabajo preciso, el que debe tener autonomía, autoridad y tecnología apropiada que le permitan realizar adecuadamente su trabajo
- *Mantener el ímpetu*: Es tarea del gestor de proyectos brindar a su equipo de desarrollo incentivos para mantener los eventuales fracasos del personal lo más reducido posible, mientras el grupo de trabajo está en la obligación de resaltar la calidad de su producto en cada una de las tareas realizadas. Por su parte, los gestores ejecutivos tienen que evitar en lo posible estar cerca del equipo de desarrollo
- *Rastrear el progreso*: Conforme se van construyendo (ej.: modelos, códigos, casos de prueba) y aprobando los productos de software (por revisiones técnicas formales) se rastrean los progresos realizados por el equipo de desarrollo; esto como parte de las actividades que aseguran la calidad del producto desarrollado. También es posible reunir y aplicar procesos y medidas del proyecto con la finalidad de evaluar el progreso en el desarrollo frente a pronósticos determinados por el empresa que realiza el desarrollo
- *Tomar decisiones inteligentes*: Decisiones tomadas tanto por gestor de proyecto como por miembros del equipo de desarrollo tienen que

dirigirse a mantener simple el proyecto. En cuanto exista la oportunidad, será decisión del gestor el uso de software comercial previamente desarrollado o componentes de software que ya existan, el uso de interfaces estándares, identificar y evitar riesgos que son evidentes, además de asignar mayor cantidad de tiempo a tareas consideradas como riesgosas

- *Realizar análisis de resultados*: Es necesario determinar la forma de extraer lo aprendido de cada proyecto, evaluar la planificación real y prevista del proyecto y las métricas del proyecto para proveer al equipo de la retroalimentación necesaria

De lo anotado en líneas anteriores se puede concluir que para la gestión de un proyecto de software es indiscutible que el gestor del proyecto y su equipo de desarrollo analicen la planificación real de los requerimientos del mismo y se formulen los objetivos claros a conseguir para evitar los fracasos durante el desarrollo, con la finalidad de garantizar la calidad del producto ofertado. Para lograr este cometido, el gestor del proyecto debe escoger el equipo idóneo para conformar su equipo de trabajo, el que deberá ser cuidadosamente elegido y al que se le deberá ofrecer los incentivos adecuados de acuerdo a sus capacidades y trabajo. Para evitar los posibles fracasos de un proyecto, la actitud de todos los involucrados en el desarrollo es de fundamental importancia, ya que gestor, grupo de equipo de trabajo y gestores administrativos deben colaborar en conjunto para la obtención de un producto de calidad.

1.1.5 Gestión de cambios

Las transiciones o cambios en el desarrollo de un programa no son extraños para el grupo de ingenieros de software. Estas transiciones provocan un sentimiento de confusión entre quienes se encuentran trabajando en un proyecto específico, la misma que resulta cuando tareas como el análisis antes

de su aplicación no se registran o no existe un control de los beneficios en que los cambios beneficiarán a la calidad y disminuirán el error (Pressman, 2005, pág. 796).

El anteriormente citado autor (Pressman, 2005, pág. 796) señala que la gestión del cambio, conocida “como *gestión de la configuración del software* GSC, es una actividad protectora (sombrija) que se aplica a lo largo del proceso de software”. Estas actividades se las realiza para (Pressman, 2005, pág. 796):

- “Identificar el cambio
- Controlar el cambio
- Garantizar que el cambio se implementará de manera adecuada
- Reportar los cambios a otros que pudieran estar interesados”

Cuando se ha finalizado el proceso de desarrollo de software, se obtiene información y se clasifica en tres categorías (Pressman, 2005, pág. 797):

- Programas de computadora (fuentes y ejecutables)
- Productos de trabajo que describen los programas de computadora (para técnicos y usuarios)
- Datos (internos o externos al programa)

Por tanto, la gestión de la configuración del software son todas las actividades que se llevan a cabo con la finalidad de gestionar el cambio durante toda la vida del software. Se estima que es una “actividad de aseguramiento de la calidad del software que se aplica a lo largo del proceso respectivo” (Pressman, 2005, pág. 798).

En consecuencia, la gestión de control de cambios para el software, son las distintas actividades que los ingenieros de desarrollo realizan para

garantizar la calidad de los productos de software, de manera que se llegue a controlar los errores.

1.1.5.1 Control de versiones

Según Sommerville (2004, pág. 636) “una versión de un sistema es una instancia de un sistema que difiere, de alguna manera, de otras instancias. Las nuevas versiones de un sistema tienen diferente funcionalidad, mejor rendimiento o incorporan reparaciones de los fallos del sistema”.

Si se piensa en crear una especial versión de un sistema, es necesario determinar cuáles son las versiones de todos los componentes que tienen que incluirse en dicho componente. Un sistema grande posee abundantes componentes de software, con varias versiones diferentes cada uno. Es necesario determinar una forma que no sean confusas para poder identificar cada una de las versiones de los componentes (Sommerville, 2004, pág. 636).

El autor antes citado (Sommerville, 2004, pág. 636) señala que hay tres técnicas básicas para identificar los componentes de software:

- *Numeración de las versiones*: se designa un número al componente de versión explícito y único. Esquema más utilizado
- *Identificación basada en atributos*: los componentes del software poseen un nombre y un grupo de atributos para cada una de las versiones del componente. Esto quiere decir que los componentes se identifican por su nombre y por los valores de los atributos
- *Identificación orientada al cambio*: cada sistema se nombra a partir de los atributos, pero también se asocia con una o más solicitudes de cambios (Sommerville, 2004, pág. 636)

Como conclusión, se entiende como versión de un sistema, a una nueva adaptación del mismo, en el cual se ha mejorado la funcionalidad del sistema y

se han reparado errores. Las versiones para identificar los componentes del software deben tener una numeración para que no existan confusiones en su identificación.

CAPÍTULO II: MERCADO LOCAL DE SOFTWARE

2.1 ¿Qué es la AESOFT?

La Asociación Ecuatoriana de Software, AESOFT, es una organización gremial privada sin fines de lucro creada en mayo de 1995 en Quito, Ecuador, que busca aglutinar a las empresas de la industria de tecnologías de información y comunicaciones propendiendo a alcanzar el desarrollo tecnológico del país. Agrupa a empresas productoras, distribuidoras y desarrolladoras de software así como compañías dedicadas a la prestación de servicios informáticos relacionados con el software y la tecnología (AESOFT, 2011).

La AESOFT es miembro de la Asociación Latinoamericana de Entidades de Tecnología Informática – ALETI. Miembro de WITSA, la Asociación Mundial de Tecnología. Miembro de la Alianza para el Emprendimiento e Innovación, AEI (AESOFT, 2011).

En AESOFT se conoce que la del software, es una de las industrias de mayor crecimiento en el mundo, que demanda mano de obra calificada, que agrega alto valor. Sus aplicaciones y servicios contribuyen sustancialmente a alcanzar (AESOFT, 2011):

- El desarrollo del comercio y de la industria en general
- La eficiencia de todos los sectores y sus productos y servicios
- El mejoramiento de la calidad de la educación y la salud pública
- El progreso de la ciencia y la tecnología (AESOFT, 2011)

2.1.1 Misión

“Somos una organización privada, sin fines de lucro, que fomenta el crecimiento del sector software en Ecuador, mediante acciones coordinadas

entre empresarios, academia y gobierno para impulsar el desarrollo nacional basado en Tecnologías de la Información” (AESOFT, 2011)

2.1.2 Visión

Asociación que lidera, representa y promueve el crecimiento y la innovación del Sector a nivel nacional (AESOFT, 2011)

2.1.3 Objetivos

Los objetivos declarados son (AESOFT, 2011):

- Fortalecer conocimientos, experiencias, profesionalización, relaciones y de ser posible, apalancar las posibilidades de negocios de las empresas nacionales de software
- Trabajar en conjunto con todas las fuerzas del Ecuador para consolidar una infraestructura de tecnología de información más fuerte y sustentable en el país, que facilite su utilización por todo tipo de usuario y masifique el acceso, brinde valor a través de la innovación, propicie un claro mapa para el desarrollo a futuro y apoye sus objetivos estratégicos de desarrollo (AESOFT, 2011)

2.1.4 Importancia de la industria.

Las Tecnologías de la Información y más propiamente el Software es transversal a todos los sectores (AESOFT, 2011):

- Genera competitividad para todos los sectores
- Es generadora intensiva de empleo
- Es generadora de valor e impacto con nivel de ganancias tempranas
- Tiene la capacidad de atraer inversiones
- Demanda recurso humano de alto valor agregado

- Se ajusta a los requerimientos del mercado (Sector Público y Privado) y produce soluciones que aportan valor (AESOFT, 2011)

2.1.5 Datos estadísticos

De acuerdo con datos proporcionados por esta institución, existen 265 empresas de software, de las cuales hay registradas en AESOFT: 77, con una facturación total de la industria por año, como sigue (AESOFT, 2011):

- 2005: US\$ 62 millones
- 2006: US\$ 99 millones
- 2007: US\$130 millones (AESOFT, 2011)

2.1.6 Políticas de la AESOFT

Esta institución tiene plenamente establecidas políticas para su funcionamiento y regulación de sus asociados (AESOFT, 2011):

- Políticas públicas y marcos regulatorios (compras públicas, comercio electrónico, fomento al software, etc.) orientados a favorecer el crecimiento del sector
- Generación de partidas arancelarias para bienes intangibles
- Políticas específicas para el sector de Tics que permitan: una significativa disminución de costos de telecomunicaciones (internet, telefonía), disminuir los costos de adquisición de insumos (computadores, partes, etc.). Políticas que incentivarán el desarrollo del Sector
- Comercializadora internacional de software de Ecuador
- Incentivos fiscales y de seguridad social
- Fuentes de financiamiento como capital de riesgo, fideicomisos, etc.
- Participación en las zonas francas (AESOFT, 2011):

2.2 Código de ética de la AESOFT

Como toda institución, la AESOFT (2011), tiene su Código de Ética, a través del cual se rigen las empresas a ella asociadas (Ver anexo 1).

2.3 Repercusión del uso de las tecnologías de la información y la comunicación Tic's en el desarrollo de la economía nacional.

En los tiempos actuales, la noción de que existe relación directa entre el desarrollo de las Tics se encuentra en relación directa con el desarrollo económico y social de un país, es cada vez más habitual tanto entre organismos internacionales “Multilaterales de Cooperación y Asistencia Técnica” (AESOFT, 2011), como empresas privadas alrededor del mundo. Dichos organismos consideran que si una nación hace el adecuado uso de las TIC's en todas las áreas de la economía, ésta podría tornarse clara y ser eficiente para el desarrollo del país.

Según señala la AESOFT (2011), durante la conferencia de las Naciones Unidas “Trade and Development (UNCTAD) en el mes de octubre del año 2010, se concluyó que, tanto los bienes como los servicios que se encuentran relacionados con las TIC's, proporcionan una serie de oportunidades a los sectores menos favorecidos y apremian a los gobiernos en vías de desarrollo a constituir el sector de las TIC's en la pieza fundamental para la planificación de sus estrategias para reducir la pobreza. Dentro del mismo contexto, se conoce que las TIC's son un medio valioso que colabora con los distintos países en la consecución de importantes progresos y logros encaminados a los Objetivos del Milenio (ODM) a cumplirse hasta el 2015, sobre todo para los países en desarrollo.

Por su parte, la misma fuente señala que datos proporcionados por el Banco Mundial indican un exagerado crecimiento en el área de las TIC's en la última década, en las economías en desarrollo. A pesar de este inusual

incremento, las economías de dichos países deben hacer frente y superar las dificultades que se presentan, como lo es la desigualdad en el acceso de las TIC's en todos los estratos sociales.

Esta fuente además manifiesta que, testimonios suministrados por el Estudio competitivo efectuado en el año 2010, empresarios ecuatorianos consideran que el país no tiene un nivel de tecnología acorde a los nuevos adelantos tecnológicos, aunque se están dando grandes pasos para conseguirlo. No obstante, dichos empresarios supieron definir como estratégico en el Ecuador el sector de tecnología, que no tiene relación con actividades petroleras, y que se encuentra en segundo lugar luego del turismo.

2.4 Mercado de software en Ecuador

Si se trata de clasificar el mercado del software, la Asociación Ecuatoriana de Software (AESOFT, 2011), segmenta el mercado de la siguiente manera:

1) Software:

- Global Application Software
- Systems Software
- Home Entertainment Software

2) Servicios de Tecnologías de información (TI)

- Data processing y Outsourcing
- Global IT Consulting and Other services

3) Software y servicios para internet

- Internet software (desarrollo y ventas)
- Online databases y servicios interactivos

4) Servicio de registro de direcciones web

- Internet design services.

2.5 Ingresos por venta de software en Ecuador

De acuerdo a datos de la AESOFT, en el país la venta de productos relacionados con el software llegó a \$ 260 millones de dólares en el año 2009. Este sector ha tenido una importante evolución en los últimos cinco años

Tabla 4: Exportaciones de software en dólares

| Cód. CIU | Actividad | 2004 | 2005 | 2006 | 2007 | 2008 | 2009 |
|-----------------|--|------------------|------------------|------------------|-------------------|-------------------|-------------------|
| K72200001 | Consultores en tipos y configuraciones de equipos informáticos y aplicación de programas informáticos conexos: análisis de necesidades y problemas de usuarios y presentación de soluciones convenientes sobre el equipo de informática. | | 7,778 | | 82,008 | 430,025 | 3,990 |
| K72200001 | Análisis, diseño y programación de sistemas. Esto supone análisis de necesidades y problemas de usuarios, para determinar la solución más económica y la producción de programas para lograr esa solución. | 1,322.271 | 786,966 | 3,541.469 | 2,590.767 | 21,659.988 | 25,270.017 |
| K72200101 | Elaboración de programas con arreglo a instrucciones de usuarios: formulación, producción y suministro de programas encargados por los usuarios, incluida documentación pertinente y programas listos para ser utilizados (programas comerciales o de fácil confección). | | | 3,700 | 30,144 | 31,000 | |
| K723000 | Procesamiento o tabulación de datos del cliente y preparación de informes en base a resultados de dicho procesamiento, incluye teclado y otro tipo de entrada de datos, conversión (tarjetas o cintas), reconocimiento óptico de caracteres, etc. | 7,055.739 | 7,298.320 | 5,267.423 | 5,758.208 | 5,711.791 | 6,493.897 |
| K72300101 | Administración y manejo permanentes de equipo de procesamiento de datos instalado por los usuarios | 186.638 | 254.566 | 283.722 | 354,924 | 423,860 | |
| K72500101 | Mantenimiento y reparación de maquinaria de informática y equipo periférico conexo. | 269.263 | 339.670 | 614.446 | 900.366 | 599.100 | 373.335 |
| K72900001 | Actividades de informática no especificadas en otra parte | 3,375 | 1.809 | | 311.212 | 325.785 | 215.444 |
| TOTAL | | 8,837.287 | 8,689.109 | 9,710.761 | 10,027.628 | 31,181.628 | 32,356.683 |

Elaborador por: la autora
Fuente: AESFOT (2011)

2.6 Listado de empresas de desarrollo de software según la AESOFT

De acuerdo con información proporcionada por la Coordinación de la AESOFT, matriz en Quito, el 27 de noviembre de 2013, las empresas de Desarrollo de Software en la ciudad de Guayaquil afiliadas, hasta la fecha, son:

Imagen 5: Listado de empresas asociadas a la AESOFT

Asociados Pertenecientes a la Ciudad de Guayaquil

| <i>Nombre</i> | <i>Teléfono</i> | <i>Email</i> | <i>Ciudad</i> | <i>Sitio Web</i> |
|--|--|--|-------------------|----------------------|
| BANRED | (02) 2523333 / (04) 2523333 | pnarvaez@banred.fin.ec | Quito, Guayaquil | www.banred.fin.ec |
| ROUTE SOLUTIONS | (04) 2295330 - 2288679 -02394001 | david.duenas@route.com.ec | Guayaquil | www.route.com.es |
| TROPIDATOS S.A. | 593 04 2203142 - 593 04 2201177 | avico@tropidatos.com | Guayaquil | www.tropidatos.com |
| BIROBID S.A. | 5103222-2643846 | jquispillo@birobid.com | Guayaquil | www.birobid.com |
| TECH NET C.A. | 042201913 | despinoza@technetca.com | Guayaquil | WWW.TECHNETCA.COM |
| EXODO | 097869995/ 04-6019190. | foardoava@exodo.com.ec | Guayaquil | www.exodo.com.ec |
| WINNERCORP S.A. | 4-5000007 4-5000008 | elias.sevilano@winnercorp.info.ec | GUAYAQUIL | www.winnerplanet.net |
| C-ENERGYTEK S.A.-SINERGIA SOLUCIONES DE SOFTWARE | 5114449 | miguel.viejo@sinergias.com | Guayaquil | www.sinergias.com |
| IMAGE TECH | 2523721-2523722-2523709 | leonardo@imagef.com / cecilia@imagef.com | Guayaquil | www.imagef.com |
| MEDIA LOGIC S.A. | 045117256/ 042280143/042280109 | LROMANZ@SIAPRE.COM | GUAYAQUIL | WWW.SIAPRE.COM |
| AGROSOFT S.A. | 04 2881333/04 2610303/ 04 2610129/0998150444 c | xcardenas@agrosoft.com.ec | Guayaquil | www.agrosoft.com.ec |
| CIMA IT | +593 4 2280217 | lcastro@cimait.com.ec | Guayaquil | www.cimait.com.ec |
| EIKON S.A. | (593-4) 5024990/ 5024993 Ext. 111 | xriera@eikon.com.ec | Guayaquil | www.eikon.com.ec |
| ALTA TECNOLOGIA EN SOFTWARE S.A. | 04 2289898 - 2289171 - 2287305 | rsanchez@altecssoft.net | Guayaquil | www.altecssoft.com |
| SIPECOM S.A. | 2561513 - 2561514 | svincoes@specom.com | Guayaquil | www.sipcom.com |
| DESARROLLO INTEGRAL DE SOLUCIONES EMPRESARIALES DIRES S.A. | (593) 4 2598292 / 3731390 EXT 1701 | fausto.ruiz@grupodifare.com | Guayaquil | www.dires.com.ec |
| GEEKS ECUADOR; MODERZACORP S.A. | 5113417 / 091214085 / | adrian.bajana@geeks.com.ec | Guayaquil | http://geeks.com.ec/ |
| TFASE S.A. | 04 2810801/ | kgabor@tfase.com | Duran y Guayaquil | www.tfase.com |

2.7 Descripción de las empresas asociadas a la AESOFT



Nombre: IROUTE SOLUTIONS

Dirección: Ciudadela Albatros calle Gaviota 115 y Pinzón

Teléfonos: (593 4) 229 5330/ 228 8679/ 239 4001

Ciudad: Guayaquil

www.iroute.com.es

DESCRIPCIÓN

i-Route Solutions empresa consultora especializada en Business Intelligence y Data Warehouse ofrece a sus clientes aplicaciones de información gerencial (para distintos tipos de negocio como Manufactura, Comercializadora, etc.) pre-construidas, personalizaciones y herramientas tecnológicas en plataformas como Oracle, Microsoft y Open Source, apoyadas con metodologías de mejoramiento empresarial como Balanced Scorecard.



Nombre: TOPIDATOS S.A.

Dirección: Av. Carlos Julio Arosemena km. 4 ½

Teléfonos: (593 4) 220 1052

Ciudad: Guayaquil

www.tropidatos.com

DESCRIPCIÓN:

Tropidatos S.A. fue creada en 1981, nos dedicamos a la comercialización de nuestro ERP ZEUS-2000 orientado básicamente a la actividad de concesionarios automotrices, pero que también cubre las necesidades de las empresas dedicadas a otras actividades comerciales.

Desarrollamos soluciones a medida, brindamos servicios de consultoría en TI y licenciamiento de productos Oracle; somos partner de Oracle desde 1999. Entregamos a nuestros clientes productos y servicios con la mejor calidad técnica y funcional. Contamos con profesionales de primer nivel, con un gran conocimiento técnico y sobre todo con una vasta experiencia en el desarrollo e implementación de soluciones informáticas, lograda a través de muchos años de fructífera labor.



Nombre: BiroBID S.A.

Dirección: Av. Agustín Freire y Av. Francisco de Orellana (Cdla. Herradura Mz. 8 Solar 12 OF. 4)

Teléfonos: (593 4) 264 3846/ 510 3222

Ciudad: Guayaquil

www.birobid.com

DESCRIPCIÓN

Es una empresa de Soluciones tecnológicas basado en la informática, fundada en el 2006 y que gracias a la calidad de su servicio en pocos años tiene clientes a nivel nacional, posee y desarrolla software para todo tipo de funcionalidad o gestión comercial, consta de un grupo de Ingenieros en Sistemas, Analistas de Sistemas y Técnicos de Hardware para desarrollar software y dar soporte a todos sus clientes.

Ofrecemos ADM DIANA, software comercial con las opciones básicas de venta, cobro, proveedores y contabilidad en línea, cumple con el REOC del SRI; ADM GOLD, software 100% orientado a empresa comerciales y sobre todos empresas distribuidoras de consume masivo, donde existe un número significativo de clientes y vendedores; ADM PLATINUM, versión más avanzada del ADM GOLD posee dos módulos más que son MOMINAS Y TECNOLOGIA MOVIL E INGRESO DE PEDIDOS POR LA WEB el módulo de equipos móviles trabaja con equipos POCKET PC y para este equipo poseemos un software llamado ADM-POCKET que le permite al vendedor tomar pedidos en sitio.



Nombre: TECHNET C.A.

Dirección: Av. Carlos Julio Arosemena Km. 2 ½ vía a Daule. C.C. Albán Borja
Puerta 2, piso 1 oficina 116

Teléfonos: (593 4) 220 1913

Fax: (593 4) 220 1913 ext. 5

Ciudad: Guayaquil

www.technetca.com

DESCRIPCIÓN

TechNet C.A. es una empresa con más de 18 años de presencia en el mercado ecuatoriano dedicada a proveer servicios, equipos y programas que cumplan con el objetivo de constituirse en tecnología eficaz y productiva para todos sus clientes, logrando así su crecimiento y satisfacción.

Buscamos convertirnos en un verdadero aliado de nuestros clientes en el aspecto de tecnología informática y sistemas de información; como parte de esta creencia, buscamos y valoramos relaciones comerciales a largo plazo.



Nombre: WINNERCORP S.A. (ZETASOFTWARE.COM ECUADOR)
Dirección: *Guayaquil:* Av. Boyacá 1630 y 10 de Agosto, piso 7, oficina 8
Teléfonos: (593 4) 223 2770/ 500 0007/8
Ciudad: Guayaquil y Quito

www.winnerplanet.com

DESCRIPCIÓN

Winnercorp es una empresa ecuatoriana especializada en Cloud Computing, contando con soluciones contables y administrativas full web. Está conformado por un equipo multidisciplinario de profesionales de tecnología de la información, que cuentan con amplia experiencia en países de la región y ponen a disposición todo su potencial al servicio de sus clientes. Los productos y servicios que evaluamos, seleccionamos y comercializamos están orientados a dar soluciones prácticas a empresas de todo tamaño y volumen que buscan automatizar su gestión comercial, administrativa, financiera y contable. Nos ocupamos de ofrecer soluciones de fácil uso, seguras y de ágil desempeño, orientadas a dar respuesta a las interrogantes que día a día hacen los empresarios.



Nombre: C-ENERGYTEK S.A.

Dirección: Alborada 12ava etapa calles Rodolfo Baquerizo Nazur y Gabriel Roldós.

Teléfonos: (593 4) 5114449

Ciudad: Guayaquil

www.cenergytek.com

SinergiaSS es una empresa dedicada al desarrollo de software a la medida en las líneas de 1) Páginas Web Transaccionales (para Internet o Intranet). 2) Desarrollo de Aplicaciones Móviles. Tablets o Celulares. Y diferentes tecnologías: Android, IOS, Blackberry 3) Desarrollo de Aplicaciones para Facebook como estrategias de incremento de los Fans 4) Consultoría y Desarrollo en Marketing Digital: Diseño y Desarrollo de Sitios Web, Diseño y Desarrollo de Fan Page de Facebook y Diseño de Twitter para presencia digital de marca. Optimizador en buscadores SEO y SEM. Consultoría sobre publicidad con Anuncios Patrocinados en Facebook. Consultoría sobre Community Manager.



Nombre: IMAGE TECH

Dirección: Av. 9 de Octubre 109 y Malecón Simón Bolívar Edif. Santistevan
Piso 3 Oficina 1

Teléfonos: (593 4) 252 3709/ 252 3721/ 252 3722

Ciudad: Guayaquil

www.imaget.com

DESCRIPCIÓN:

Imagetech genera soluciones en tecnología de información alineadas a la estrategia de tu negocio ayudándote a cumplir tus objetivos más ambiciosos. Te integramos a la tecnología digital.



Nombre: MEDIALOGIC S.A.

Dirección: Cdla. Kennedy Vieja, Av. San Jorge y Calle F 2da Oeste # 117

Teléfonos: 2280143/5117255/2280109

Ciudad: Guayaquil

www.medialogic.com.ec

DESCRIPCIÓN:

Somos un grupo de expertos profesionales con sólidos y amplios conocimientos en el desarrollo de programas informáticos para cubrir con las necesidades y expectativas de nuestros clientes.

Nuestra experiencia en el campo del software a medida y en el de los sistemas, nos permite ofrecer a nuestros clientes un servicio integral, que incluya el mantenimiento de su sistema informático, además de cualquier necesidad de software a medida



Nombre: AGROSOFT S.A.

Dirección: Ciudadela Urdesa Central, Guayacanes 303 y Calle Tercera

Teléfonos: (593 4) 288 1333 / 261 0303

Fax: (593 4) 261 0303 ext. 109

Ciudad: Guayaquil

www.agrosoft.com.ec

DESCRIPCIÓN

Agrosoft S.A. es una empresa que se dedica al desarrollo e implementación de sistema computación (software) con soluciones específicas para empresas comerciales, agrícolas y de servicios. Cuenta con 15 años de experiencia brindando soluciones tecnológicas a más de 300 empresas privadas u ONG locales con exportaciones de productos y servicios al sector bananero. Entre sus principales productos tenemos: **Xass 11:** Software Administrativo-Financiero Empresarial con NIIF. **Banaxass 7:** Software agrícola y gerencial de producción de banano. **Exportxass 3:** Software de proceso de operación y exportación. Entre sus principales servicios tenemos consultorías y capacitaciones.



Nombre: CIMA IT

Dirección: Cdla. Kennedy Norte, Av. Miguel H. Alcívar y Eleodoro Arboleda, Edificio Plaza Center, piso 8, oficina 802-803

Teléfonos: (593 4) 228 0217

Ciudad: Guayaquil

www.cimait.com.ec

DESCRIPCIÓN:

CIMA IT, empresa consultora con sede en Ecuador y con inicios estadounidenses y mexicanos hacia el año de 1997, nace dedicada a proveer servicios especializados y de excelencia en Tecnología de Información, brindando soluciones que den soporte a empresas de gran robustez. CIMA IT ofrece consultoría especializada en ERP's Oracle Peoplesoft y Oracle JDEdwards, además de tecnologías integradas de alta confiabilidad desarrolladas por especialistas en nuestra Fábrica de Software y que ayudarán a clientes como usted a satisfacer rápida y efectivamente sus necesidades. Somos una firma especializada en Inteligencia de Negocios con Oracle Business Intelligence.



Nombre: EIKON

Dirección: Av. Francisco Boloña 715 y Calle Décima

Teléfonos: (593 4) 228 0217

Ciudad: Guayaquil

www.eikon.com.ec

DESCRIPCIÓN:

Eikon S.A., es una empresa de desarrollo de software especialista en Firma Electrónica & PKI y Comercio Exterior, así como prestadores de Servicios de Confianza Digital. Nos orientamos a diferentes mercados que tiene una necesidad común, la de obtener productos de software con calidad y de alto rendimiento que garanticen su inversión tecnológica.

Tenemos 12 años de vida como empresa, con más 150 clientes a nivel nacional, y con nuestras alianzas con IPSCA de España y STS Group de Francia más de 8000 clientes a nivel mundial, tiempo durante el cual nos hemos focalizado en la integración de soluciones. Los principales ejecutivos de nuestra compañía cuentan con más de 16 años de experiencia en procesos de Ingeniería de software.

Actualmente, tenemos 4 áreas de negocio muy bien definidas:

Servicios de Confianza Digital (Confidence Suite), Aplicaciones para empresas de comercio exterior (Productos Nathalia), Soluciones de Firma Electrónica y PKI, Desarrollo de Software a Empresarial a Medida.



Nombre: ALTA TECNOLOGÍA EN SOFTWARE ALTECSOFT S.A.

Dirección:

Teléfonos: 593-4-2289898 * 593-4-2287305 * 593-4-2289171

Ciudad: Guayaquil

www.dires.com.ec

DESCRIPCIÓN:

Con experiencia en el mercado de desarrollo y comercialización de software administrativo y de gestión. Está integrada por un equipo de profesionales altamente capacitados en el diseño, construcción, capacitación e implementación de sistemas de información. Las aplicaciones administrativas con más de diez años de trabajo continuo orientado hacia el mercado de empresas usuarias de sistemas administrativos o que se inician en el proceso de sistematización de la información que producen sus diferentes áreas.

ALTECSOFT ha desarrollado el negocio de software administrativo clasificando su producto en tres líneas: **ALTECSOFT Enterprise** para medianas empresas; **ALTECSOFT Small-Business** para pequeñas empresas y **ALTECSOFT Lite** para empresas o negocios personales (microempresas). Cada línea de producto tiene asociado los diferentes servicios de Asesoría, Capacitación y Soporte.



Nombre: SIPECOM

Dirección: Pedro Carbo 123 y Junín, Edificio Pasaje Comercial La Merced, 2do piso, oficina 2

Teléfonos: (593 4) 256 1513

Ciudad: Guayaquil

www.sipecom.com

DESCRIPCIÓN:

Cuenta con más de 11 años brindando servicios y productos en el área de tecnología. Especializada en el Desarrollo de Software, OutSourcing en Desarrollo y Mantenimiento de Software, Consultorías de Bases de Datos, Inteligencia de Negocios y Capacitación. Es una compañía ecuatoriana con profesionales, capacitados y certificados, asegurando la mejor solución a la medida de sus clientes. Contamos con talento humano con más de 20 años de experiencia, desarrollando proyectos de software.



Nombre: DIRES

Dirección: Av. Carlos Julio Arosemena km. 2 ½ y Las Monjas

Teléfonos: (593 4) 2598292

Ciudad: Guayaquil

www.dires.com.ec

DESCRIPCIÓN:

Desarrollo Integral de Soluciones Empresariales DIRES S.A. es una empresa ecuatoriana con más de 16 años de experiencia en la provisión de Tecnología y Servicios para el sector de la Salud.

Fundada en 1995, actualmente es líder en brindar soluciones tecnológicas, asesoría y servicios para el sector farmacéutico ecuatoriano. Socio de negocios Microsoft en el segmento ISV Solutions desde el año 2004 y calificado como IBM Business Partner desde el 2011, DIRES cuenta con un portafolio de productos y servicios acorde con las necesidades del mercado y los avances de la tecnología.

DIRES S.A. ha desarrollado soluciones que son líderes en el mercado ecuatoriano, logrado una participación de mercado de más del 40% en el sector de la salud. Soluciones como Neptuno y GestoRx nos permiten alcanzar a más de 2,400 clientes en 160 ciudades en el Ecuador y explorar territorios internacionales con proyecto implementados exitosamente en Colombia, Venezuela y República Dominicana.



Nombre: GEEKS Marketing Interactivo
Dirección: Kennedy Norte Mz. 407 Solar 1
Teléfonos: (593 4) 5113417
Ciudad: Guayaquil

www.geeks.com.ec

DESCRIPCIÓN:

En el mundo digital todo avanza muy rápido, tendencia que se ve reflejada en la publicidad, en las herramientas digitales y en lenguajes de programación.

Somos una agencia joven que lleva cuatro años, pero que conoce muy bien a quienes les hablamos, por eso, tenemos una consigna: mantenernos actualizados para conseguir resultados según los objetivos de nuestros clientes.

CAPÍTULO III: METODOLOGÍA

3.1 Enfoque metodológico

Al entrar en materia de metodología, es necesario precisar que este estudio es una investigación y por lo tanto “la investigación es un conjunto de procesos sistemáticos, críticos y empíricos que se aplican al estudio de un fenómeno” (Hernández Sampieri et al., 2010, pág. 4).

El enfoque propuesto que permitirá cumplir objetivos y responder preguntas de investigación está basado en la combinación de los enfoques cuantitativo y cualitativo de la investigación los que, de acuerdo a Hernández Sampieri et al., (2010, pág. 4) hacen uso de cuidadosos procesos metódicos y empíricos como medida para generar conocimiento. Utilizan, de forma general, cinco fases que son similares y están relacionadas entre sí:

- Observación y evaluación de fenómenos
- Se fijan teorías o ideas resultantes de la observación y valoración de los fenómenos
- Argumentan el fundamento de las suposiciones o ideas
- Verifican esas suposiciones o ideas a través de las pruebas o del análisis
- Plantean nuevas observaciones y valoraciones para que se aclaren, transformen y fundamenten las teorías e ideas o para generar otras nuevas

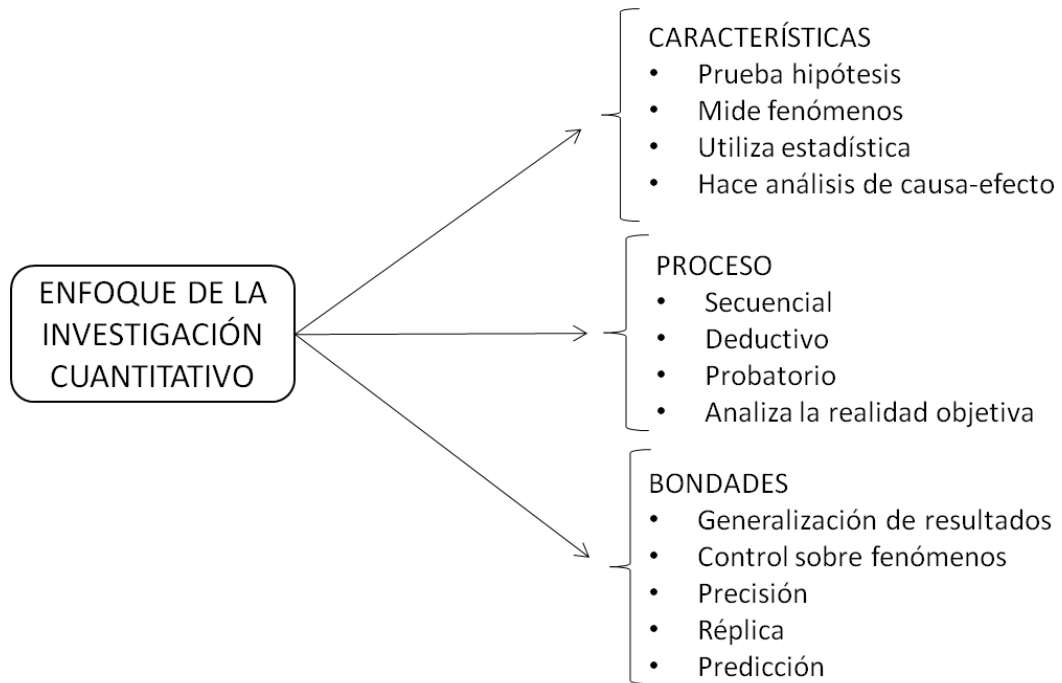
Si se habla del *enfoque metodológico cuantitativo* propiamente dicho, el mismo autor señala algunas características (Hernández Sampieri et al., 2010, pág. 5-6):

1. Se propone un tema de estudio delimitado y concreto, del cual se derivan las preguntas de investigación

2. Luego de plantear el problema de estudio, se toma en consideración la verificación de la literatura para construir un marco teórico que guiará el estudio. Planteará una o varias hipótesis sobre los temas que va a examinar para determinar si son ciertos o no y las somete a prueba a través de los diseños de investigación adecuados. Si se confirman las hipótesis o el resultado es coherente con éstas, se aporta evidencia en su favor; por el contrario, si son impugnados, se eliminan con la finalidad de buscar otras explicaciones e hipótesis. Si las hipótesis son aceptadas, entonces se confía en la teoría en que éstas se fundamentan, de lo contrario se procede a descartar las hipótesis y, de cierta forma, la teoría
3. De este modo se generan las hipótesis antes de proceder a recepcionar y analizar los datos
4. La recolección de los datos tiene como base la medición, esto es, se miden las variables o conceptos contenidos en las hipótesis. Como el enfoque cuantitativo pretende medir, los hechos a estudiar deben poder observarse o referirse en el “mundo real”
5. Los datos resultan de mediciones y son analizados por métodos estadísticos
6. Durante el proceso de medición es necesario controlar que otras posibles definiciones existentes, diferentes a la propuesta del estudio se descarten, se prescinda la indecisión y reduzca el error. De este modo se cree en la experimentación o también en las pruebas de causa-efecto
7. La interpretación de los datos cuantitativos se la realiza en base a las hipótesis iniciales y análisis previos realizados, constituidos por la teoría. La interpretación es la explicación de los resultados que se ajustan en el conocimiento que existe

8. El enfoque cuantitativo debe ser imparcial. Los datos y circunstancias estudiados y/o medidos no tienen que afectarse por la persona que investiga. Ésta debe tratar de que todo aspecto que pueda afectar los resultados del estudio o interferir en los procesos estén presentes en el momento de la recolección y análisis
9. El proceso del enfoque cuantitativo sigue un esquema predecible y estructurado y es indispensable que toda decisión crucial sea plantada antes de la recolección de la información
10. El enfoque cuantitativo busca generalizar los resultados que se obtienen en la muestra, a un universo o población. Del mismo modo, se trata de que los estudios realizados se repliquen.
11. Con este enfoque se trata de explicar y pronosticar los hechos que se investigan, tratando de encontrar regularidades y relaciones causales entre elementos. Lo que significa que el objetivo principal es construir y demostrar teorías que explican y predicen
12. En este enfoque los datos que se generan son válidos y confiables, y las conclusiones obtenidas permitirán la generación de conocimiento
13. El enfoque cuantitativo hace uso de la lógica o deducción que parte de la teoría. De ésta resultan expresiones lógicas o hipótesis que el investigador busca probar
14. El enfoque metodológico cuantitativo procura determinar leyes universales y causales
15. La búsqueda de datos e información cuantitativa se presenta en la realidad que rodea al individuo. Con esto se llega a una definición de la concepción de la realidad

Gráfico 5: Enfoque cuantitativo de la investigación

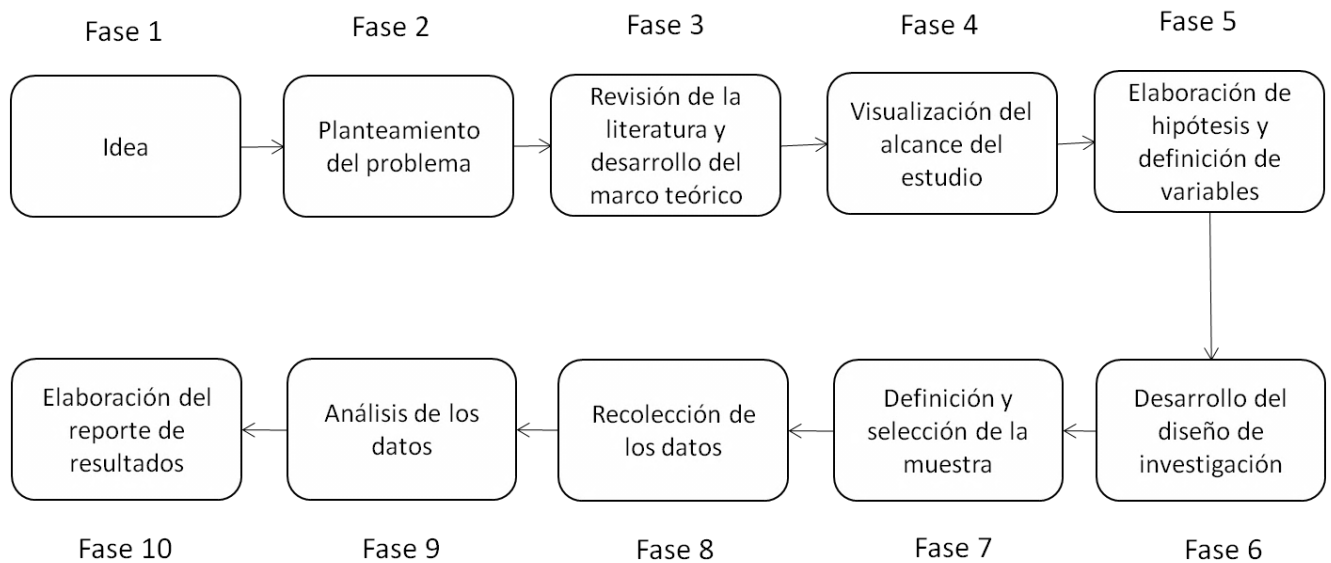


Elaborado por: la autora

Fuente: Hernández Sampieri (2010)

El gráfico 5 permite visualizar de forma sintética algunas características, el proceso y las bondades del enfoque cuantitativo.

Gráfico 6: Fases del proceso cuantitativo



Elaborado por: la autora

Fuente: Hernández Sampieri (2010)

El gráfico 6 señala paso a paso las fases que cumple la metodología de investigación cuantitativa.

En cuanto al *enfoque cualitativo*, Hernández Sampieri et al., (2010, pág. 9-10) señala las siguientes características:

1. Se sugiere un tema de estudio que no se encamina por un proceso definido. Las propuestas no son tan definidas como lo presenta el proceso cuantitativo y las preguntas de investigación no necesariamente ha sido definidas
2. Se comienza con una investigación de la realidad social con la finalidad de desarrollar una teoría que se relacione con los datos, de acuerdo con lo que observa y poder determinar qué es lo que ocurre. El fundamento de la investigación cualitativa es una lógica y proceso inductivo (primero exploración y descripción, posteriormente

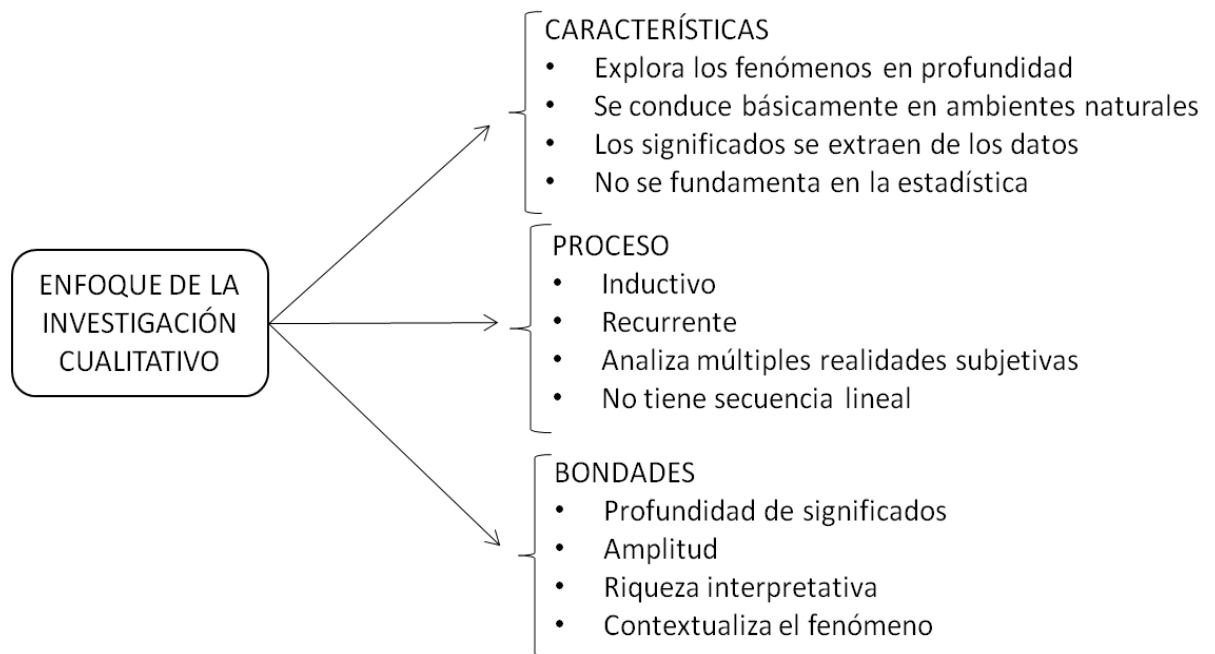
forjar perspectivas teóricas). Es un proceso que va de lo particular a lo general

3. La generación de las hipótesis se realizan durante el proceso del estudio cualitativo. Es por este motivo que, generalmente, no se comprueban las hipótesis y se refinan de acuerdo a la recopilación de mayor cantidad de información o resultan del estudio
4. Los métodos de recolección de datos no son generales ni predeterminados. La medición numérica no existe: el análisis no se fundamenta en la estadística. La recolección de los datos significa conocer la perspectiva y puntos de vista de los que intervienen; además son de interés las interacciones que se producen entre sujetos, agrupaciones y comunidades. Se cuestiona por temas abiertos, recoge información de datos que se manifiestan mediante lenguaje escrito, verbal, no verbal y visual
5. Se utilizan técnicas de recolección de información, “como la observación no estructurada, entrevistas abiertas, revisión de documentos, discusión en grupo, evaluación de experiencias personales, registro de historias de vida, e interacción e introspección con grupos o comunidades” (Hernández Sampieri et al., 2010, pág. 9)
6. La investigación es más flexible y se desarrolla entre las respuestas y el desarrollo de la teoría. El objetivo es rehacer la realidad, de acuerdo a cómo la observan los actores
7. Inexistencia de manipulación y estimulación de la realidad
8. Sus cimientos se encuentran en el plano de la interpretación de cómo se entiende el concepto de las acciones de seres vivos, esto es, humanos y sus instituciones
9. La realidad se la determina por intermedio de las distintas definiciones de quienes participan en el estudio en referencia a sus

propias realidades. Es así como De este modo se concentran las realidades de los participantes, del investigador y las que se suceden por interacción de todos los actores, las que se modifican durante la investigación y son las fuentes de datos

10. La investigación se ubica en la variedad de ideologías y cualidades únicas de los individuos a los que se los estudia
11. El estudio cualitativo no busca extender de forma probabilística los resultados obtenidos a grupos amplios ni tampoco conseguir muestras representativas
12. Es un grupo de prácticas para interpretar el mundo y hacerlo visible, transformarlo y convertirlo en un conjunto de representaciones (observaciones, anotaciones, grabaciones, documentos)

Gráfico 7: Enfoque cualitativo de la investigación

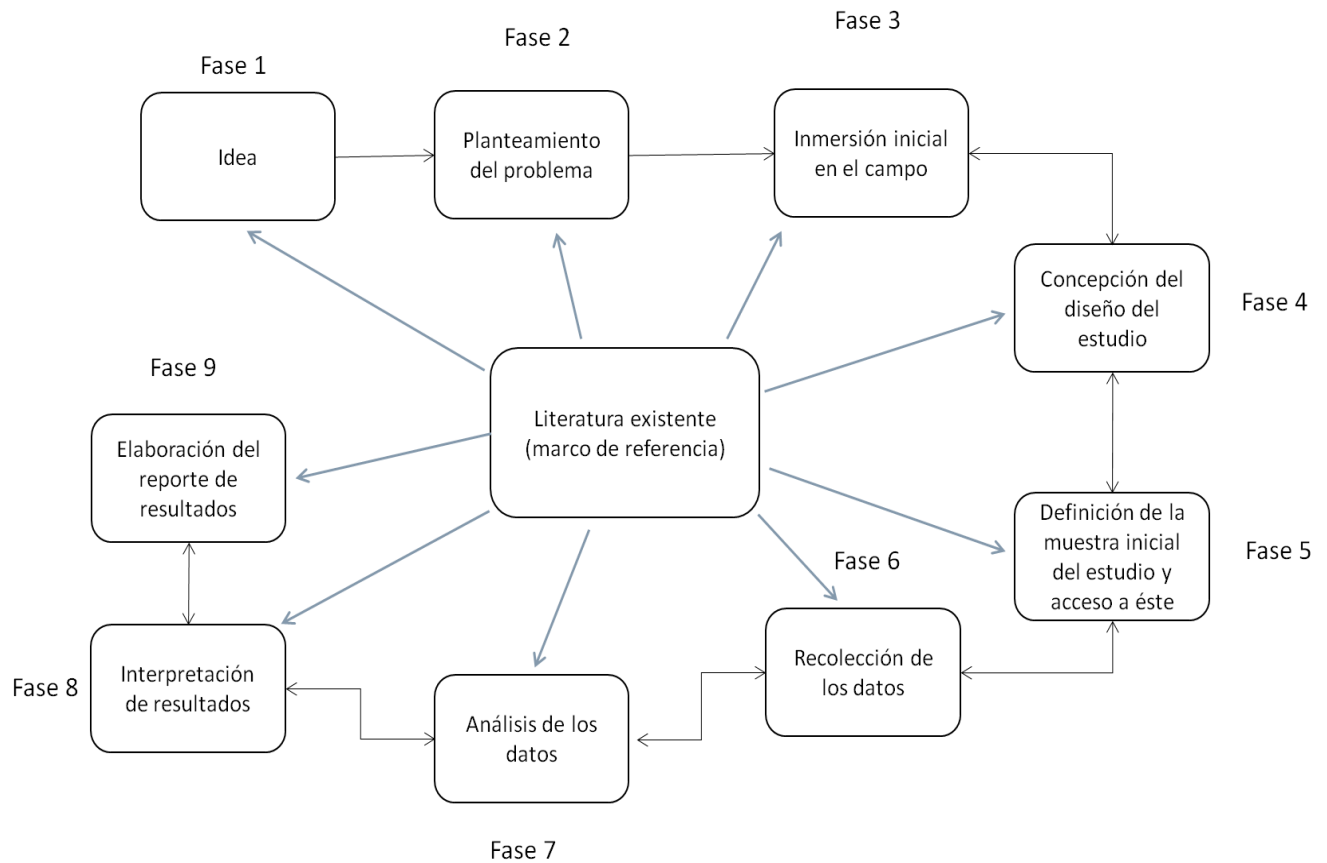


Elaborado por: la autora

Fuente: Hernández Sampieri (2010)

El gráfico 7 recoge algunos datos del enfoque metodológico cualitativo. En el mismo se aprecia características, proceso de investigación y bondades.

Gráfico 8: Fases del proceso cualitativo



Elaborado por: la autora
Fuente: Hernández Sampieri (2010)

En el gráfico 8 se detalla las fases del proceso cualitativo el que, como se aprecia de forma clara y precisa, que se fundamenta en el análisis de la literatura existente, de la cual se obtienen los datos preliminares para la investigación.

Como se dijo al inicio de este estudio, la metodología a utilizar está fundamentada en los métodos cuantitativo y cualitativo.

El método cuantitativo se lo utilizará para el análisis de los datos obtenidos de la entrevista que se aplicará a las empresas de desarrollo de software de Guayaquil asociadas a la AESOFT y otras empresas que están dedicadas a esta actividad.

El método cualitativo, por su parte, es el que proveerá de datos o estudios previos realizados anteriormente. Esta literatura existente la constituye el Estudio estadístico exploratorio de las empresas desarrolladoras de software asentadas en Guayaquil, Quito y Cuenca, realizado por la Escuela Superior Politécnica del Litoral, textos de Ingeniería de Software de donde se obtienen conceptos básicos en que se fundamenta la entrevista y el Catálogo de Soluciones de Software 2012-2013 proporcionado por la AESOFT, del que se toman aspectos en relación al estudio de mercado y datos de las empresas de desarrollo de software de Guayaquil a ella asociadas. Todas estas fuentes sirven de base para la construcción del marco teórico de este estudio.

3.2 Diseño de investigación

Hernández Sampieri et al., (2010, pág. 120-121) señala como diseño de investigación “al plan o estrategia concebida para obtener la información que se desea”.

El mencionado Hernández Sampieri et al., (2010, pág. 120-121) sugiere que en la metodología de **investigación cuantitativa**, la calidad de la investigación se relaciona con la jerarquía de aplicación del diseño tal como fue proyectado. Este diseño se debe adaptar a eventualidades o variaciones en los escenarios.

En la misma línea del autor referido, se adopta la clasificación de la metodología cuantitativa:

1. Investigación experimental:

- Pre experimentos
- Experimentos “puros”
- Cuasi experimentos

2. Investigación no experimental:

- Diseños transversales
- Diseños longitudinales

El estudio en cuestión seguiría el modelo de *investigación no experimental*, definida por Hernández Sampieri et al., (2010, pág. 149) como “estudios que se realizan sin la manipulación deliberada de variables y en los que solo se observan los fenómenos en su ambiente natural para después analizarlos”, en donde la observación de la realidad existente no es manipulada por el investigador, sino que es tomada tal y como se la encuentra; las variables independientes no se manipulan, no se tiene control sobre las mismas ni se puede influir en ellas, por cuanto ya sucedieron, lo mismo que los efectos de aquellas.

Para efectos de esta investigación, el diseño acorde a las necesidades de este estudio es el diseño de *investigación transeccional o transversal*, que recoge datos en un tiempo y momento único. Describe variables y examina su influencia e interrelación en determinado momento. Es como tomar una fotografía de algo que sucede (Hernández Sampieri et al., 2010, pág. 151).

El diseño transeccional se clasifica en tres (Hernández Sampieri et al., 2010, pág. 152):

- Exploratorios
- Descriptivos
- Correlacionales-causales

El propósito del *diseño transeccional descriptivo* según Hernández Sampieri et al., (2010, pág. 152-153) tiene como finalidad investigar la incidencia de las características o niveles de una(s) variable(s) presente(s) en una población. “El procedimiento consiste en ubicar en una o diversas variables a un grupo de personas u otros seres vivos, objetos, situaciones, contextos, fenómenos, comunidades; y así proporcionar su descripción”.

En cuanto a este tipo de diseño, cabe anotarse que se lo utilizará para establecer la incidencia de las variables presentes en la investigación, cuando se realice la entrevista a la muestra seleccionada de las empresas de desarrollo de software de Guayaquil asociadas a la AESOFT y demás empresas que se dedican a esta actividad.

En la metodología de **investigación cualitativa** no existe un diseño específico, sino que es importante la revisión de la literatura y la teoría. La literatura es de importancia para (Hernández Sampieri et al., 2010, pág. 369-370):

- Descubrir conceptos claves no tomados en cuenta de antemano
- Tener nuevas ideas sobre métodos de recolección de datos y análisis, en cuanto se refiere a su utilidad a otros investigadores
- Tener en mente los errores que otros han cometido anteriormente.
- Descubrir distintas maneras de pensar y abordar el planteamiento.
- Mejorar el entendimiento de los datos y profundizar las interpretaciones.

3.3 Objeto de estudio

Empresas de desarrollo de software de Guayaquil asociadas a la AESOFT y otras empresas que están dedicadas a esta actividad.

3.4 Población y muestra

Arias (2006, pág. 81) define como *población o población objetivo* a “un conjunto finito o infinito de elementos con características comunes para los cuales serán extensivas las conclusiones de la investigación. Ésta queda delimitada por el problema y por los objetivos del estudio” (Arias, 2006, pág. 81).

El mencionado autor (Arias, 2006, pág. 83) señala como *muestra* “un subconjunto representativo y finito que se extrae de la población accesible”.

Entre los tipos de muestra que el autor señala, se encuentran:

1. Muestreo probabilístico o aleatorio

- Muestreo al azar simple
- Muestreo al azar sistemático
- Muestreo estratificado
- Muestreo por conglomerados

2. Muestreo no probabilístico

- Casual o accidental
- Muestreo intencional u opinático
- Muestreo por cuotas

Por su parte Hernández Sampieri et al., (2010, pág. 173) señala que la *muestra* “es un subgrupo de la población de interés sobre el cual se recolectaran datos, y que tiene que delimitarse de antemano con precisión;

este deberá ser representativo de dicha población”. Se busca conseguir que los resultados hallados en la muestra se generalicen a la población (Hernández Sampieri et al., 2010, pág. 173).

Asimismo, el autor presenta la clasificación para el muestreo (Hernández Sampieri et al., 2010, pág. 171):

1. Muestreo probabilístico o aleatorio

- Muestra aleatoria simple
- Muestra estratificada
- Muestreo por racimos o clústeres

2. Muestreo no probabilístico o dirigido

Como muestra probabilística, según Hernández Sampieri et al., (2010, pág. 176), se entiende al “subgrupo de una población en el que todos los elementos de ésta tienen la misma posibilidad de ser elegidos”.

Además, una muestra probabilística se utiliza en diseños transeccionales descriptivos para medir y analizar variables en la población, por intermedio de métodos estadísticos en una muestra, donde se presupone que esta es probabilística y que cada uno de los elementos de la población puede ser escogido. Los elementos de la muestra tendrán valores parecidos a los de la población, de tal forma que cuando se realice la medición en la muestra, se proporcionarán estimados precisos del conjunto mayor. “La precisión de dichos estimados depende del error en el muestreo, que es posible calcular” (Hernández Sampieri et al., 2010, pág. 190).

De los conceptos presentados por los autores antes citados, se puede determinar que población y muestra son dos elementos indispensables para conseguir y medir datos en toda investigación. Como *población* se entiende un conjunto de elementos que constituyen un universo sobre el cual se va a

realizar una investigación, del cual se extraerá un grupo representativo de elementos para aplicar los métodos de investigación que se van a utilizar en dicho estudio. A este grupo se conoce con el nombre de *muestra*.

Se conocen algunos tipos de muestra. Para este estudio se aplica la *muestra probabilística para diseño transeccional descriptivo*.

3.5 Técnicas e instrumentos de recolección de datos

Tabla 5: Técnicas e instrumentos de recolección de datos

| | OBJETIVOS | SUB OBJETIVOS | TÉCNICAS | INSTRUMENTOS | INFORMANTE |
|---|--|---|-----------------|---------------------|-----------------------|
| 1 | Elaborar un diagnóstico sobre la existencia de empresas que desarrollan software en la ciudad de Guayaquil, su tamaño y antigüedad, las herramientas tecnológicas para prestar el servicio | Obtener listado de empresas de la AESOFT | Documentos | Documentos | AESOFT |
| | | Analizar características de las empresas | Entrevista | Tabulación de datos | Representante empresa |
| 2 | Determinar el efecto alcanzado a través de los medios y herramientas aplicables al control de gestión de proyectos orientados al desarrollo de software | Conocer las herramientas aplicadas en la gestión de proyectos | Entrevista | Guión de entrevista | Representante empresa |
| | | Analizar el uso de las herramientas | Entrevista | Tabulación de datos | |
| 3 | Determinar el nivel de confianza percibida a nivel de usuarios respecto a las empresas que desarrollan software en Guayaquil | Identificación del nivel de confianza | Entrevista | Tabulación de datos | No aplica |
| | | Obtener información de los usuarios | Entrevista | Guión de entrevista | Representante empresa |
| 4 | Presentar recomendaciones válidas y aplicables que pudieran ser puestas en consideración para comprobar la efectividad del software, utilizable tanto para las empresas que lo desarrollan así como para las que reciben este servicio | Analizar los resultados de la investigación | | | No aplica |
| | | Formular recomendaciones a seguir por las empresas | | | No aplica |

Elaborado por: la autora

3.6 Técnicas de procesamiento y análisis de datos

Las técnicas de procesamiento de datos son los procesos y operaciones a las cuales serán sometidos todos los datos provenientes de la entrevista a las empresas de desarrollo de software, como instrumento de recolección de datos. El resultado proveniente de la tabulación de los datos permitirá conseguir los resultados que den respuesta a las preguntas de investigación y a los objetivos específicos.

CAPÍTULO IV: PRESENTACIÓN DE RESULTADOS

4.1 Prueba piloto aplicada a empresas de desarrollo de software

Previo a la obtención de los resultados finales, se realizó una prueba piloto a un grupo de cinco empresas de desarrollo de software pertenecientes a la AESOFT y otras empresas dedicadas a esa actividad en la ciudad de Guayaquil.

Las preguntas del cuestionario fueron cuidadosamente elaboradas en base a la necesidad de recolectar la información que lleve al cumplimiento de los objetivos. Para ponderar los valores en la puntuación de preguntas puntuales que requieren de un puntaje, se utilizó la serie de Fibonacci por ser ésta una serie geométrica, en vista de la necesidad de buscar una puntuación que pueda permitir elegir la prioridad en la respuesta a la pregunta.

4.2 Análisis de resultados

Los siguientes son los resultados de la tabulación de las preguntas.

ENTREVISTA

DATOS INFORMATIVOS:

1. Género del entrevistado

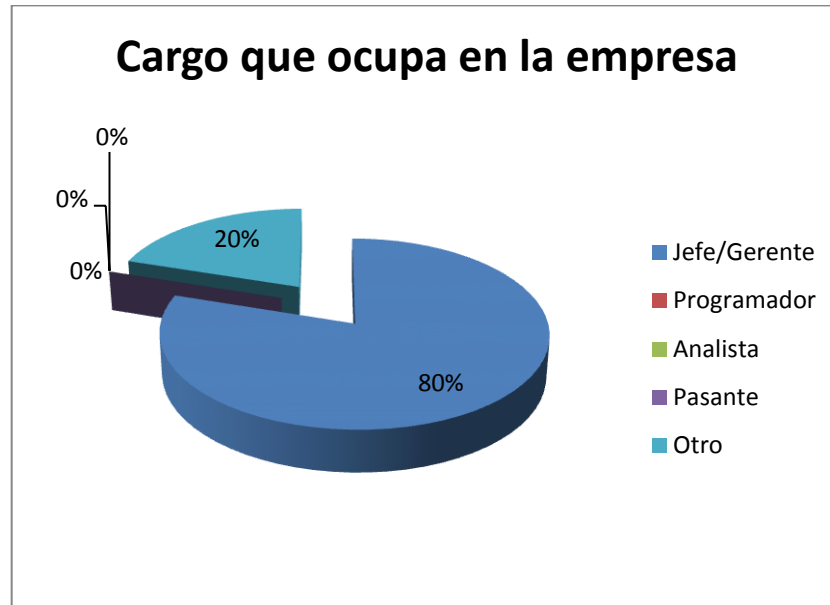
- Hombre
- Mujer



El resultado que arroja la primera pregunta, es que todas las empresas entrevistadas están dirigidas por hombres.

2. Cargo que ocupa en la empresa

- jefe/gerente de proyectos
- programador
- analista
- pasante
- otro



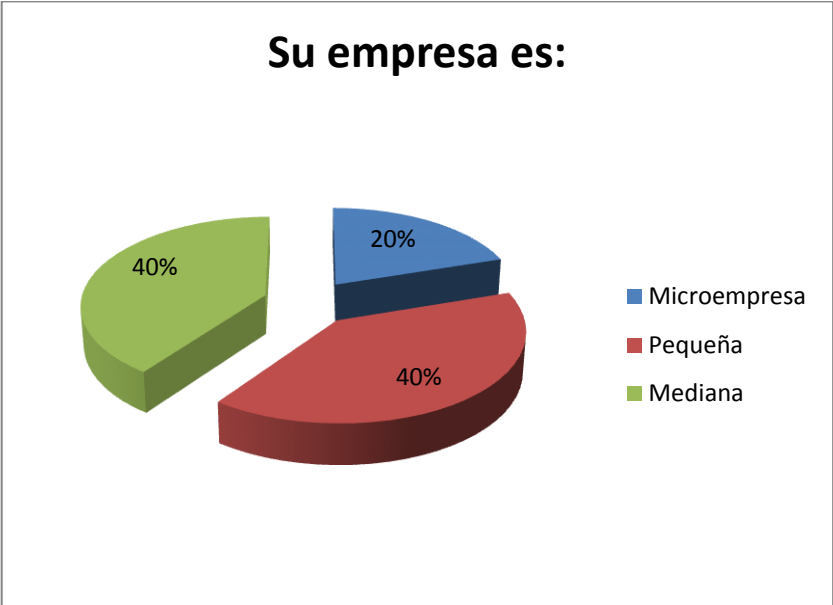
En esta segunda pregunta, el 80% de los entrevistados son Jefes o Gerentes de Proyectos, frente al 20% que señaló como “otros” la siguiente respuesta.

IDENTIFICACIÓN BÁSICA DE LA EMPRESA

3. Su empresa es:

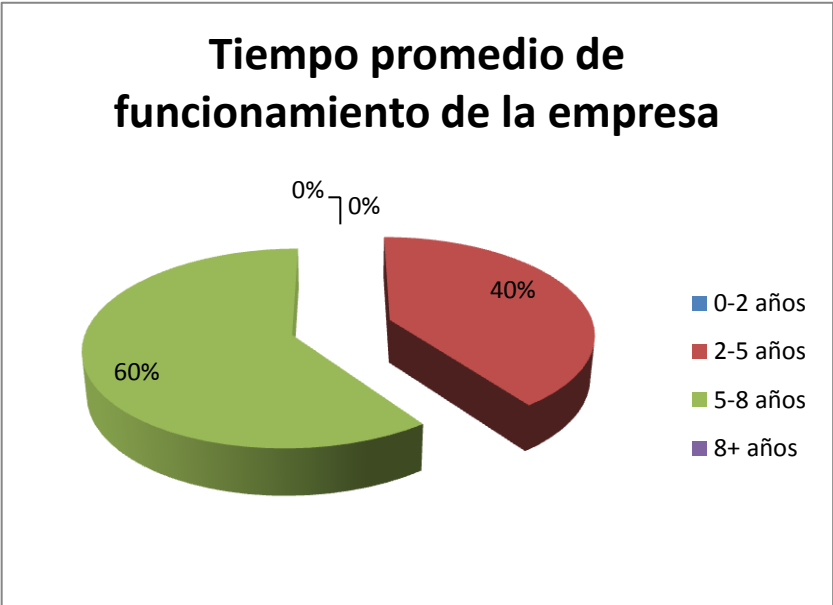
- Microempresa (1 a 9 trabajadores)
- Pequeña (10 a 49 trabajadores)
- Mediana (50 a 99 trabajadores)

Los datos obtenidos en esta pregunta, claramente permiten identificar como las opciones de mayor puntuación a la pequeña y mediana empresa, con un 40% de los encuestados cada una, como tendencia del tamaño de las empresas que desarrollan software.



4. Tiempo promedio de funcionamiento de la empresa (rango)

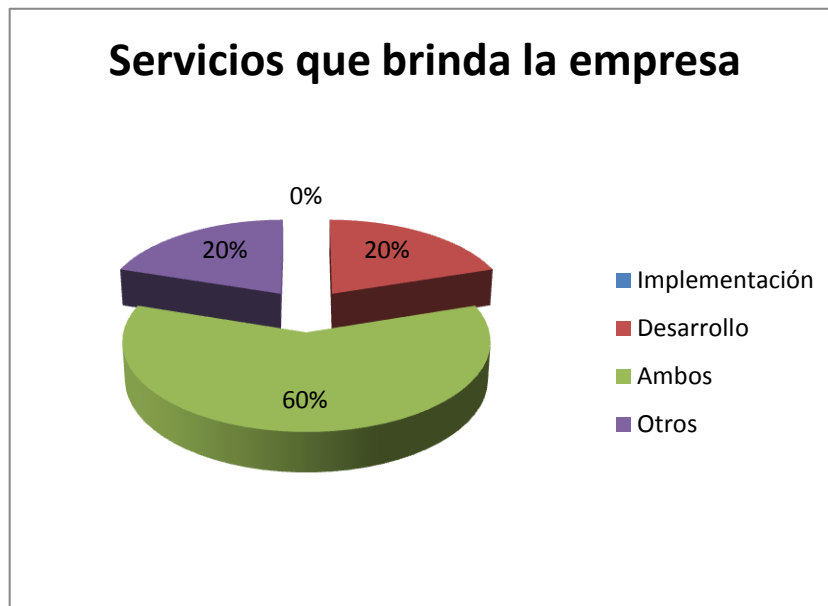
- de 0 hasta 2 años
- más de 2 hasta 5 años
- más de 5 hasta 8 años
- más de 8 años



En cuanto al tiempo promedio de funcionamiento de las empresas, el 60% tienen entre 5 y 8 años en ejercicio, frente al 40% que tienen en el mercado entre 2 y 5 años de funcionamiento, lo que indica que si existe posicionamiento y preferencias de las empresas en el mercado local de software.

5. Servicios que brinda la empresa

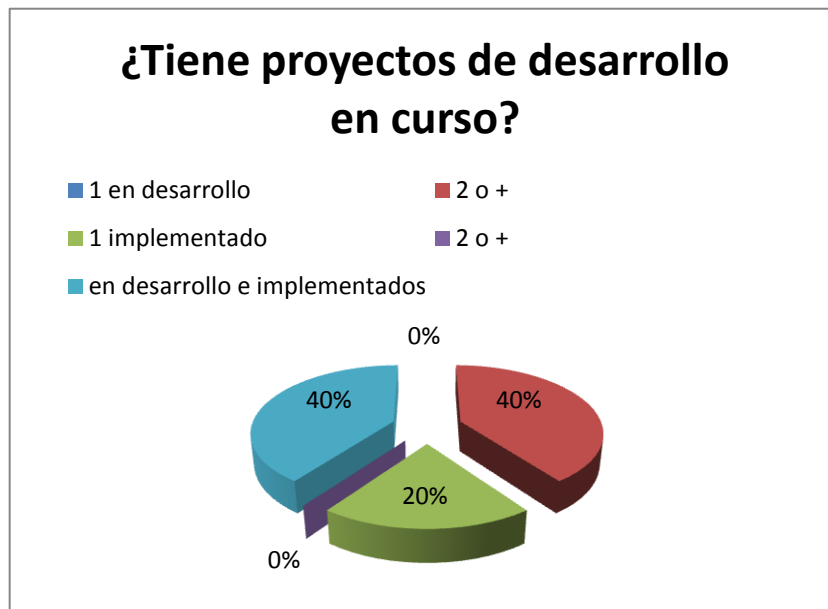
- Implementación de sistemas
- Desarrollo de sistemas
- Ambos



La tabulación de los datos de esta pregunta arroja resultados que indican un 60% de empresas que realizan desarrollo e implementación de sistemas, frente al 20% de empresas que realizan únicamente desarrollo y otro 20% que realizan otras actividades.

6. ¿Tiene proyectos de desarrollo en curso? ¿Cuántos?

- 1 proyecto en desarrollo
- 2 o más proyectos en desarrollo
- 1 proyecto implementado
- 2 o más proyectos implementados
- Proyectos en desarrollo e implementados

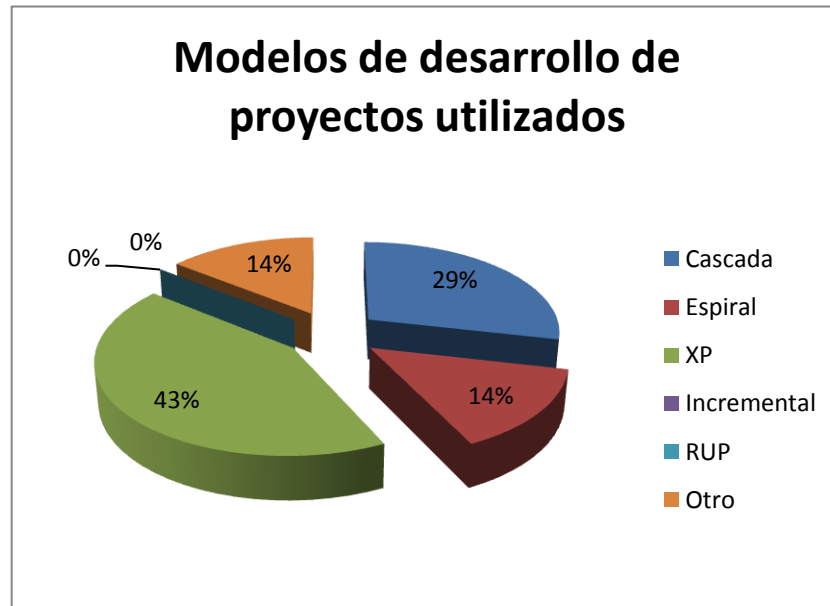


En relación con la pregunta 6, 40% de las empresas señalaron que tenían proyectos en desarrollos e implementados, junto a otro 40% que manifiesta tener 2 o más proyectos en desarrollo. Un 20% señala tener un proyecto implementado y las otras opciones se colocan en el 0%

PARTE TÉCNICA

7. Modelos de desarrollo de proyectos utilizados:

- cascada
- espiral
- programación extrema
- incremental
- rup
- otro

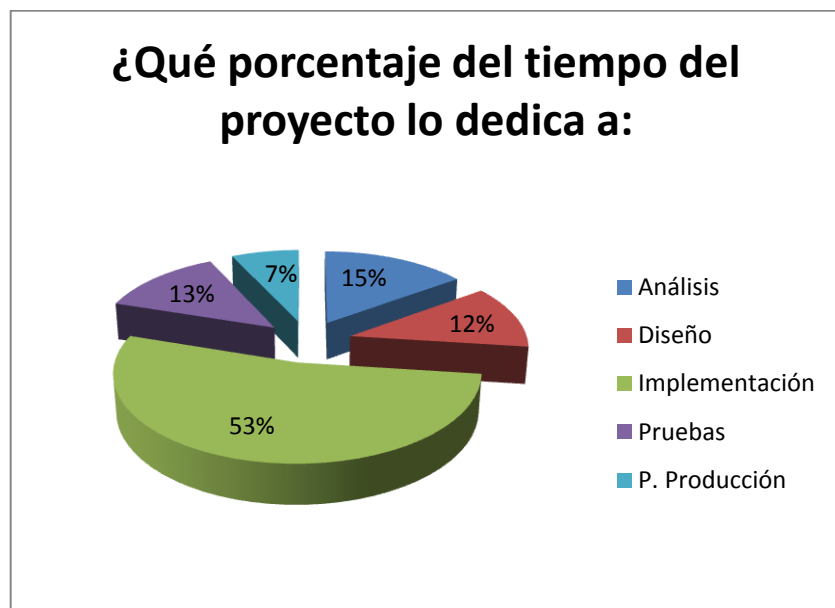


El 43% de las empresas se manifiestan con Programación Extrema como método de desarrollo de proyectos utilizado, seguido por el 29% que prefieren el método Cascada, el 14% el método Espiral y otro tipo de metodología. Las otras opciones se mantienen en el 0%

8. ¿Qué porcentaje del tiempo del proyecto lo dedica a

- análisis
- diseño
- implementación
- pruebas
- puesta en producción

Justifique la respuesta_____



Como se puede inferir fácilmente, es a la fase de implementación de un proyecto a la que más se destina tiempo del total del proyecto. Un 15% destinan el tiempo al análisis, un 13% a la fase de pruebas, un 12% al diseño, y por último un 7% de las empresas entrevistadas se pronunció por la puesta en producción.

9. ¿Qué porcentaje del costo del proyecto lo destina a

- gestor de proyecto
- analistas
- desarrolladores
- equipo de pruebas

De las 5 empresas entrevistadas, una de ellas no contestó a la pregunta, por lo tanto no se pudo realizar la tabulación.

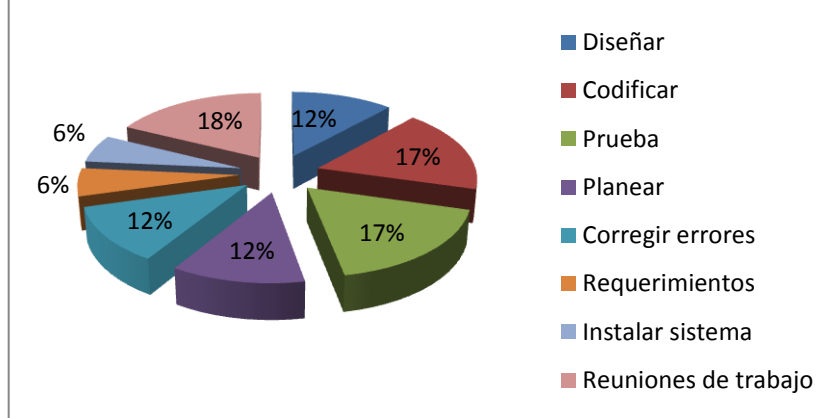
10. ¿Cómo definiría métrica de software?

De las 5 empresas entrevistadas, sólo tres respondieron a esta pregunta. Las dos restantes alegaron falta de tiempo para responder una pregunta abierta.

11. ¿Qué actividades con control por horas realiza para el desarrollo del proyecto?

- diseñar
- codificar
- probar el sistema
- planear
- corregir errores
- requerimientos
- instalar el sistema
- reuniones de trabajo

¿Qué actividades de control por horas realiza para el desarrollo del proyecto?

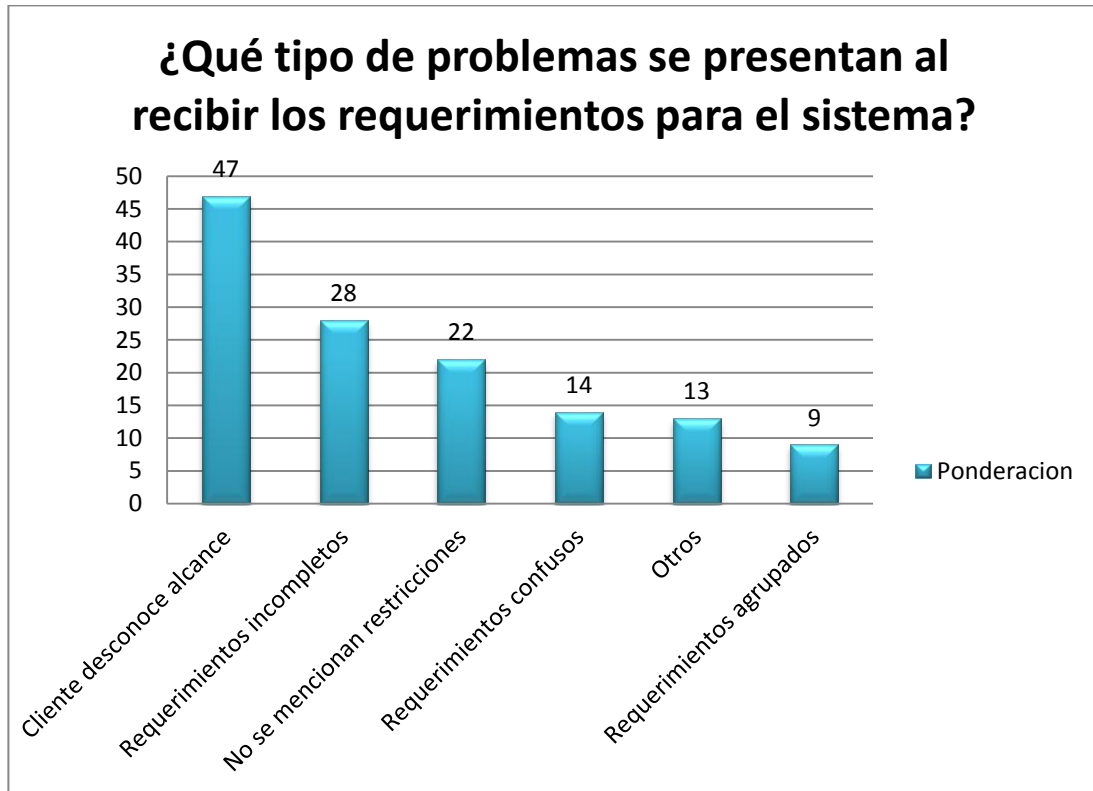


En esta pregunta, las tendencias son relativamente bajas entre las opciones y distribuidas casi de forma similar. Esto se evidencia en las opciones: Reuniones de trabajo, codificación y pruebas, que tienen el 18%, 17% y 17% en su respectivo orden. Le siguen diseño, corrección de errores y planeación, con un 12% de la distribución del tiempo cada una de las opciones mencionadas. Finalmente, análisis de requerimientos y la instalación del sistema como tal, tienen el 6% cada una de ellas.

12. ¿Qué tipo de problemas se presentan al recibir los requerimientos para el sistema? Ordene por prioridad

- cliente desconoce el alcance del proyecto
- se reciben requerimientos incompletos
- se reciben requerimientos confusos
- no se mencionan restricciones
- se reciben requerimientos agrupados
- otros

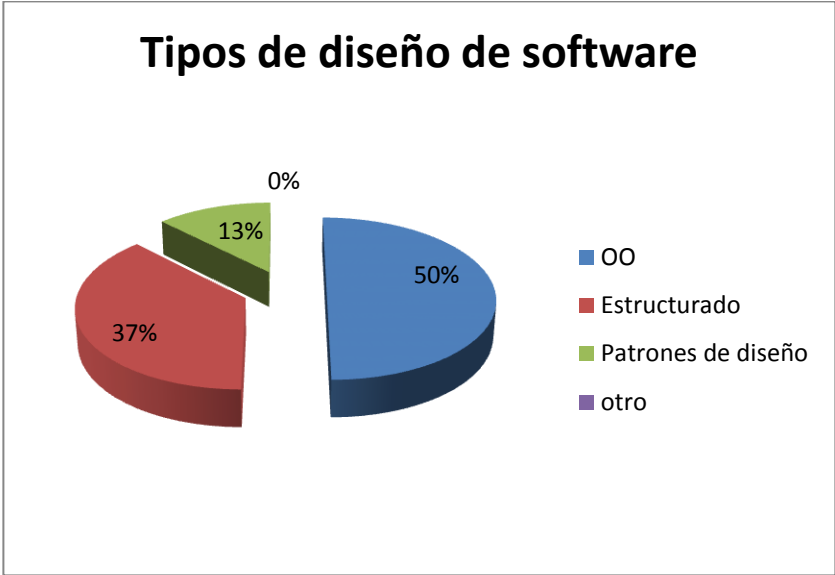
¿Cuáles? _____



La pregunta 12 es una de aquellas en la cual se utilizó ponderación para buscar una manera de dar una prioridad al orden de las opciones que son escogidas por las empresas encuestadas. De esta forma se evidencia de que la primera opción con mayor prioridad señalada por los entrevistados es: cliente desconoce el alcance del proyecto (47), seguido de que los usuarios presentan requerimientos incompletos (28), requerimientos confusos (22), no se mencionan restricciones (14), requerimientos agrupados (13) y otros (9).

13. Tipos de diseño del software

- orientado a objetos
- estructurado
- patrones de diseño
- otros



14. ¿Hace inspección en la fase de diseño?

- si
- no

En esta pregunta, el 100% de los encuestados respondió hacer inspección en la fase de diseño del proyecto.



15. ¿Hace reutilización de software?

- Si
- no
- desconoce

¿Por qué? _____



El 100% de los entrevistados hace reutilización de software.

Si la respuesta es afirmativa, en qué porcentaje estima la reutilización?

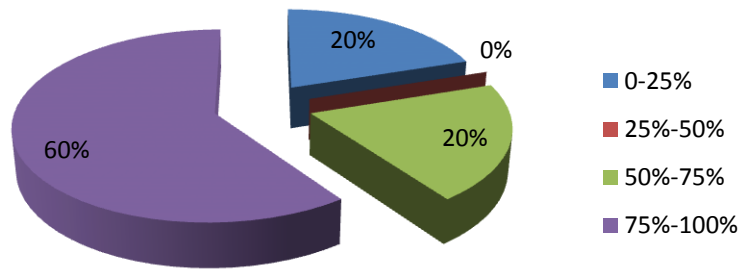
0-25%

25%-50%

50%-75%

75%-100%

¿En qué porcentaje estima la reutilización?



En cuanto al porcentaje de reutilización, el 60% afirma realizar esta actividad en un 75%-100%, un 20% lo hace en el rango 0-25% - 50-75%, en tanto que en el rango comprendido entre 25%-50% no se ubica ninguna empresa.

¿Le sirvió como estrategia? Justifique la respuesta

- 0-25%
 - 25%-50%
 - 50%-75%
 - 75%¿100%
-

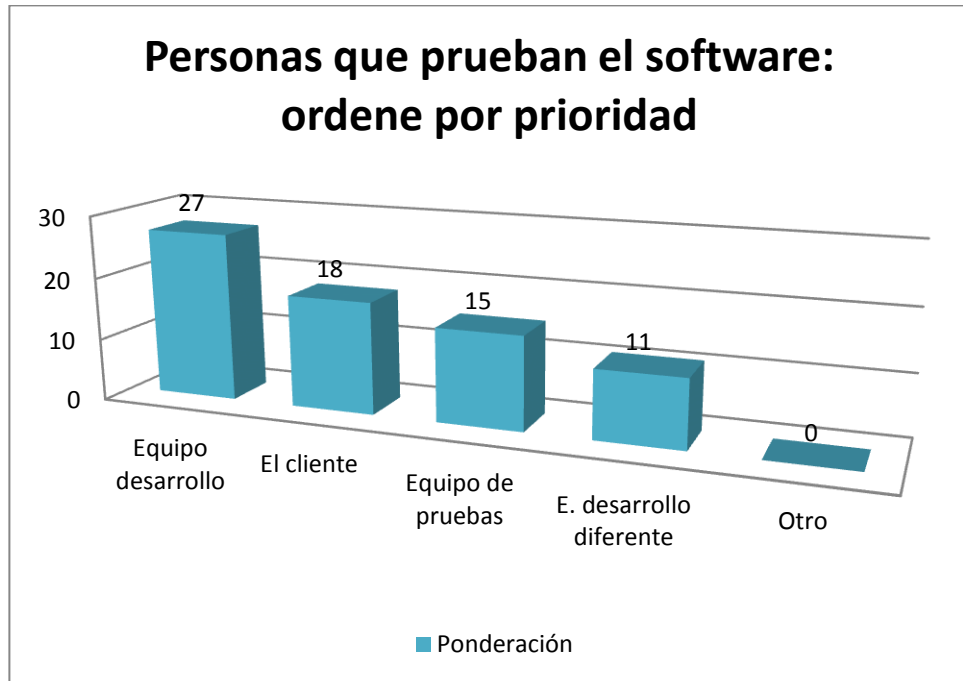


Si la reutilización de software sirvió como estrategia de desarrollo, un 60% respondió de forma afirmativa en el rango 25-50%. Para el 40% de los entrevistados le sirvió en el rango 75-100%, mientras que para el rango 75-100% no se registró ninguna empresa.

16. Personas que prueban el software. Ordene por prioridad

- el equipo de desarrollo
- un equipo de desarrollo diferente
- un equipo de pruebas
- el cliente
- otros

Especifique _____

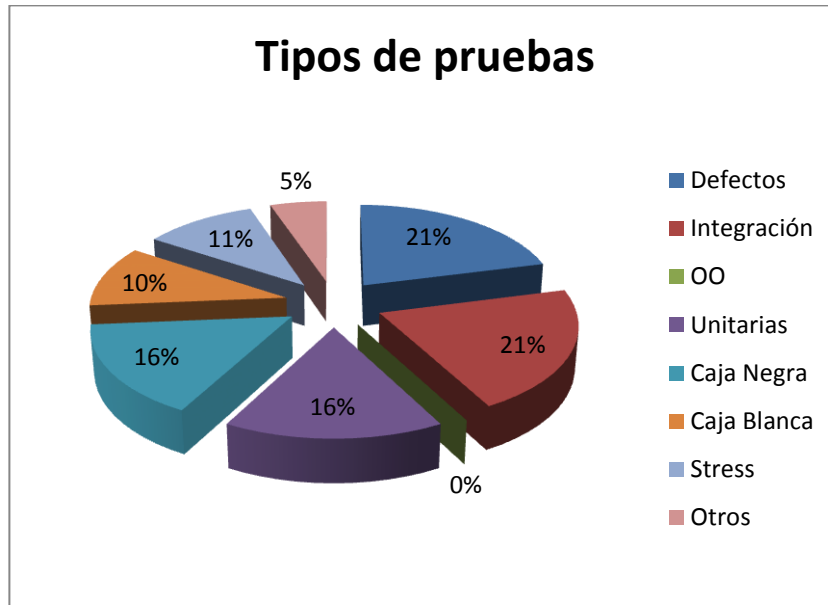


De acuerdo a los datos recibidos de las empresa y conforme a la ponderación dada a esta pregunta, se establece que el grupo de pruebas de software que más se destaca es el Equipo de desarrollo (27), seguido por el Cliente (18), el Equipo de pruebas (15) y un Equipo de desarrollo diferente (11)

17. Tipos de pruebas

- defectos
- integración
- orientado a objetos
- unitarias
- caja negra
- caja blanca
- stress
- otro

¿Cuáles? _____

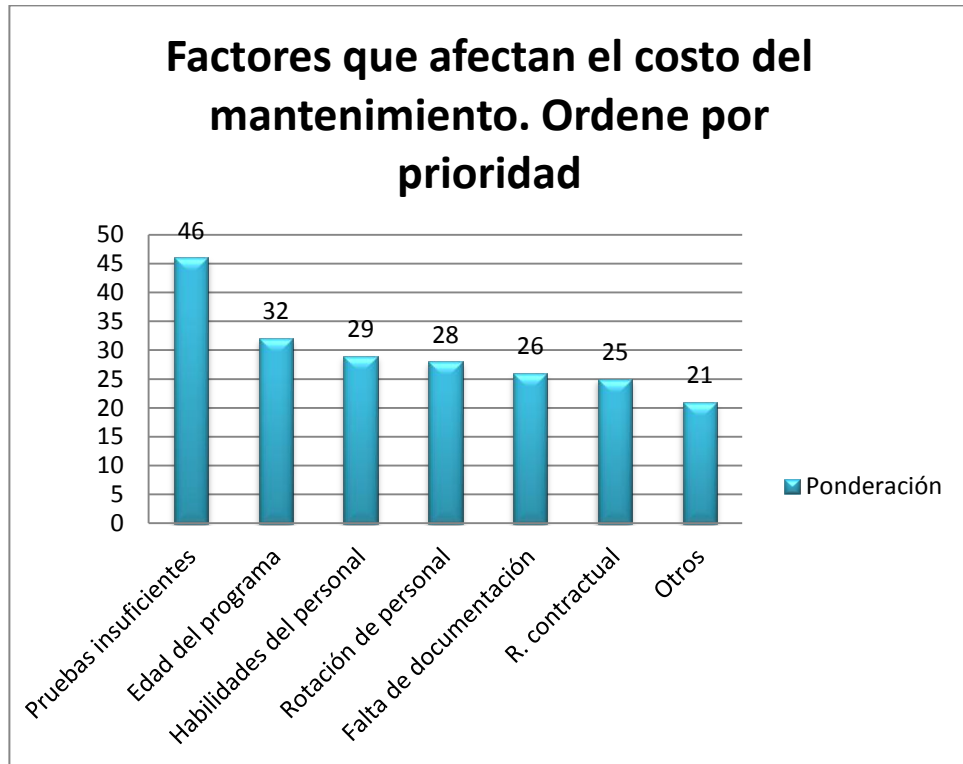


La pregunta relacionada con el tipo de pruebas arrojó resultados diversos. Con el 21% se encuentran las opciones de pruebas de integración y defectos, seguido del 16% las pruebas caja negra y unitarias. Con el 11%, 10% y 5% las pruebas stress, caja blanca y otros tipos de pruebas respectivamente.

18. Factores que afectan el costo del mantenimiento. Ordene por prioridad

- rotación de personal
- pruebas insuficientes
- responsabilidad contractual
- falta de documentación
- edad y estructura del programa
- habilidades del personal
- otros

¿Cuáles? _____



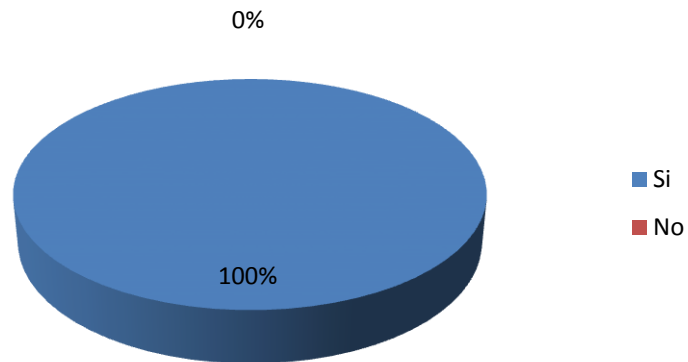
Pruebas insuficientes (13) es la opción de mayor referencia como alternativa a los factores que afectan el costo del mantenimiento. Le sigue Responsabilidad contractual (8), Falta de documentación (5), Edad y estructura del programa (3), Habilidades del personal (2) y Otros (1).

19. ¿Corrige los defectos del software al ser reportados?

- si
- no

En esta pregunta, el 100% de los entrevistados afirmó corregir los defectos del software al ser reportados por los clientes.

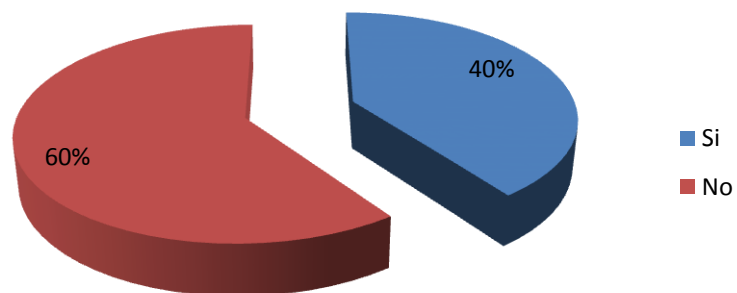
¿Corrige los defectos del software al ser reportados?



20. ¿Registra el tiempo que utiliza en corregir los defectos?

- si
- no

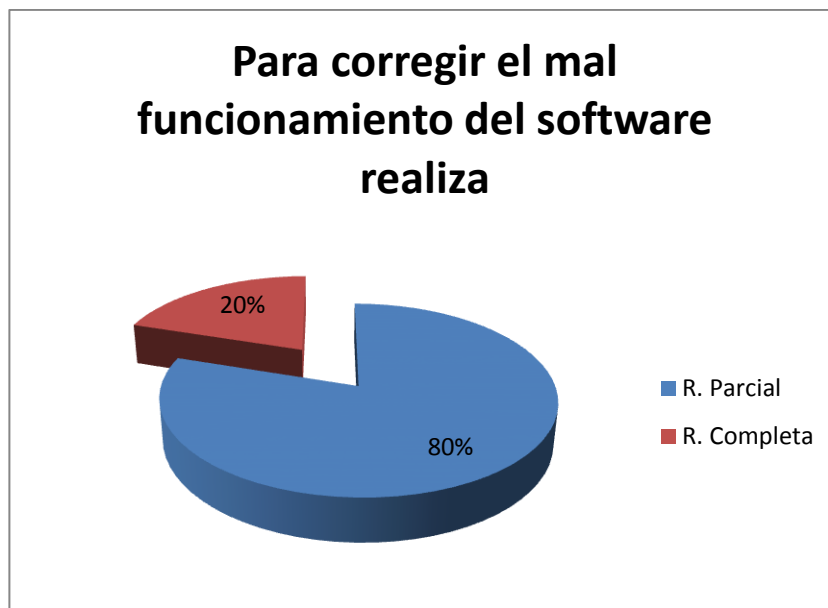
¿Registra el tiempo que utiliza en corregir los defectos?



El 60% de los entrevistados señaló que si registra el tiempo que le toma corregir errores, frente al 40% que manifiesta no hacerlo.

21. ¿Para corregir el mal funcionamiento del software realiza

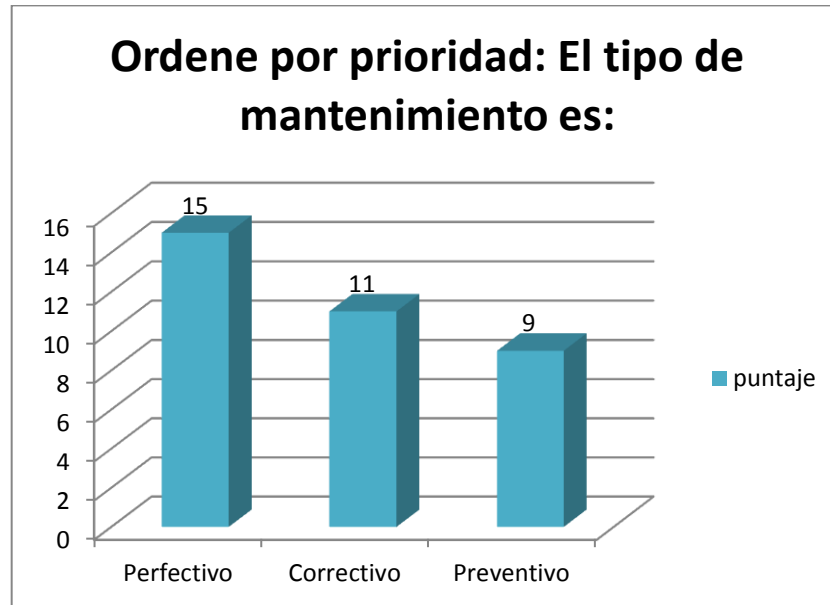
- reinstalaciones parciales
- reinstalaciones completas



En la pregunta 21, en relación con las reinstalaciones totales o parciales para corregir el mal funcionamiento del software, el 80% de los entrevistados señaló que realiza reinstalaciones parciales. El 20%, por su parte señaló que hace reinstalación completa del software.

22. Ordene por prioridad: El tipo de mantenimiento es

- perfectivo (mejora la funcionalidad del software)
- correctivo (repara las fallas en el software)
- preventivo



En lo relacionado con el tipo de mantenimiento, la opción Mantenimiento perfecto es la más escogida por las empresas (15), seguida del Mantenimiento correctivo (11) y por último el Mantenimiento preventivo (9).

Un aspecto que es importante señalar es que las preguntas que incluyen especificar o justificar una respuesta, no fueron contestadas por el entrevistado, alegando falta de tiempo.

4.3. Entrevista final a empresas de desarrollo de software

En la entrevista definitiva a las empresas de desarrollo de software que se tomó como muestra para la realización de este estudio, se modificaron las preguntas, de manera que la tabulación de los resultados sea más fiable, sin utilizar ningún sistema para ponderar datos. Las preguntas que contenían otras preguntas, fueron separadas y consideradas como nuevas.

4.4. Análisis de resultados

Se debe considerar que para la realización de este proyecto, la población escogida estuvo limitada a las empresas de desarrollo de software de Guayaquil asociadas a la AESOFT y otras empresas dedicadas a esta actividad. En vista de ser pequeña la población, la muestra seleccionada es intencional, a la cual el acceso es por intermedio de amigos o docentes que sirven de intermediarios para que éstas se lleven a cabo.

Los datos obtenidos de la entrevista final aplicada a las empresas de desarrollo de software de Guayaquil afiliadas a la AESOFT y otras empresas que se dedican a esta actividad son los siguientes:

Entrevista a empresas de desarrollo de software

FECHA: _____

DATOS INFORMATIVOS:

1. Género del entrevistado

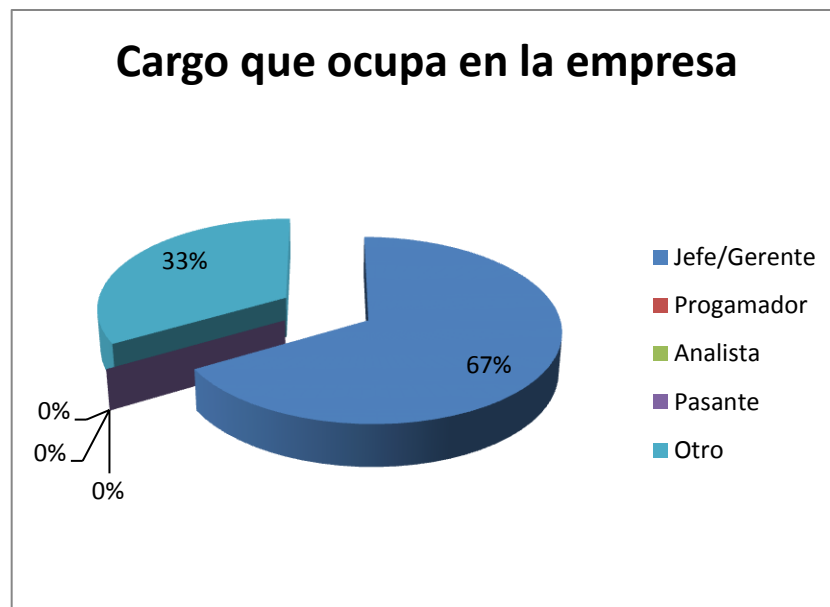
- 1) Hombre
- 2) Mujer



El 100% de las empresas entrevistadas están a cargo de hombres, como respuesta a la pregunta 1.

2. Cargo que ocupa en la empresa

- 1) jefe/gerente de proyectos
- 2) programador
- 3) analista
- 4) pasante
- 5) otro

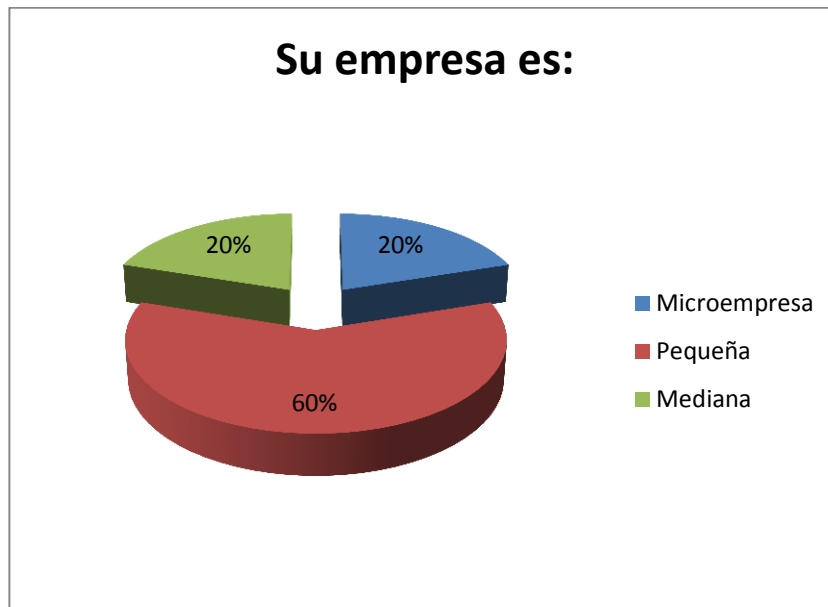


El 67% de los entrevistados señalaron que son Jefes/Gerentes de proyectos, en tanto que un 33% señala que tienen otras funciones dentro de la empresa.

IDENTIFICACIÓN BÁSICA DE LA EMPRESA

3. Su empresa es:

- 1) Microempresa (1 a 9 trabajadores)
- 2) Pequeña (10 a 49 trabajadores)
- 3) Mediana (50 a 99 trabajadores)



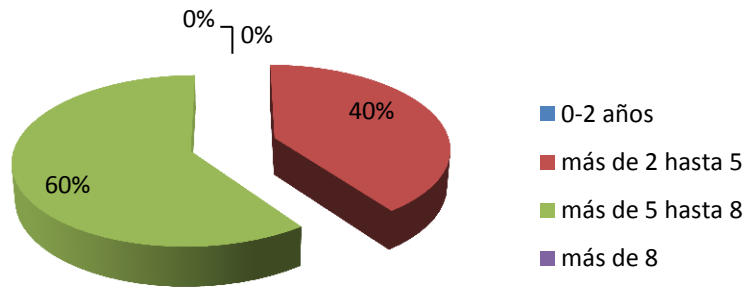
En esta pregunta se visualiza que las empresas entrevistadas se encuentran en el grupo de pequeña empresa (60%), en tanto que los valores para microempresa y mediana empresa son similares (20% cada uno).

4. Tiempo promedio de funcionamiento de la empresa (rango)

- 1) de 0 hasta 2 años
- 2) más de 2 hasta 5 años
- 3) más de 5 hasta 8 años
- 4) más de 8 años

El tiempo promedio de funcionamiento de las empresas entrevistadas se encuentra en el rango de más de 5 a 8 años (60%), seguido de otro grupo de empresas que están en funcionamiento en el rango más de 2 hasta 5 años.

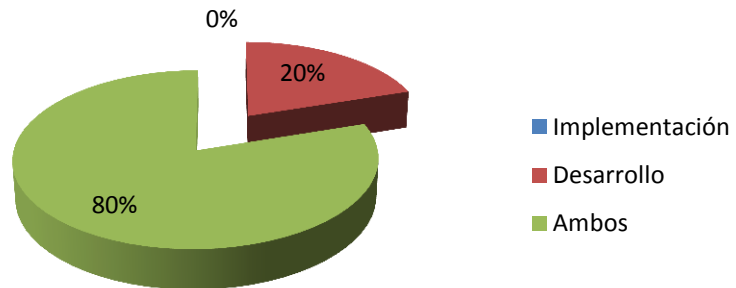
Tiempo promedio de funcionamiento de la empresa



5. Servicios que brinda la empresa

- 1) Implementación de sistemas
- 2) Desarrollo de sistemas
- 3) Ambos

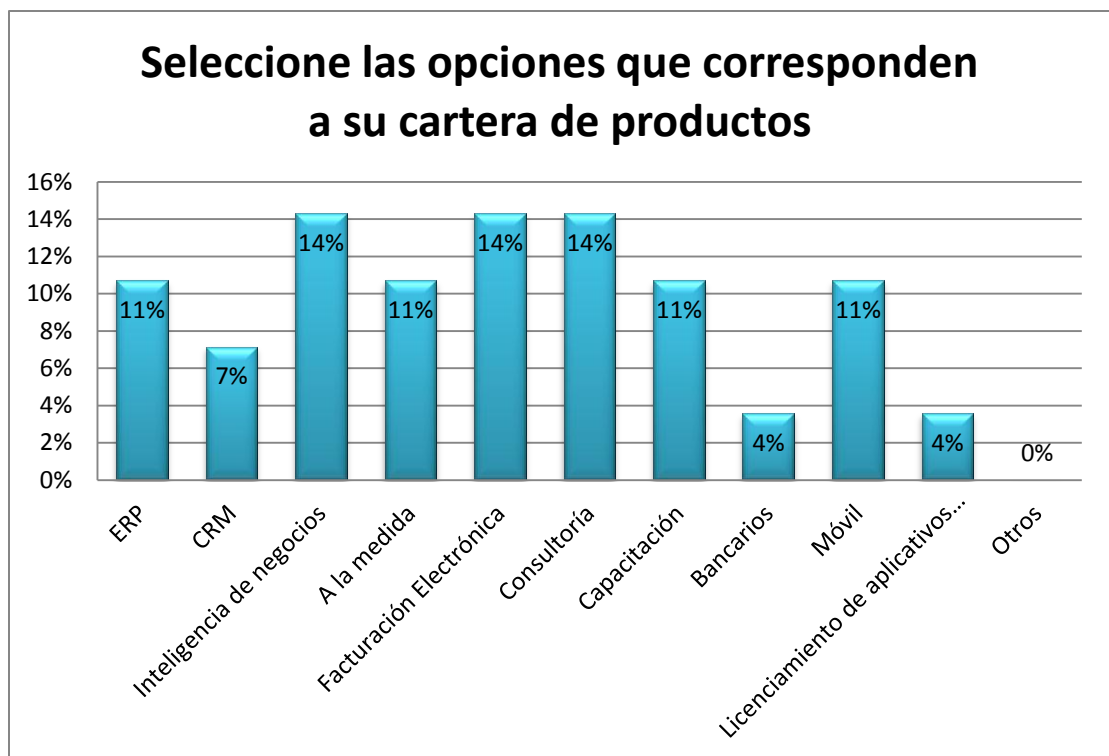
Servicios que brinda la empresa



El 80% de las empresas entrevistadas se dedica a servicios de desarrollo e implementación de sistemas. El 20%, sólo a desarrollo.

6. Selecciones las opciones que corresponden a su cartera de productos

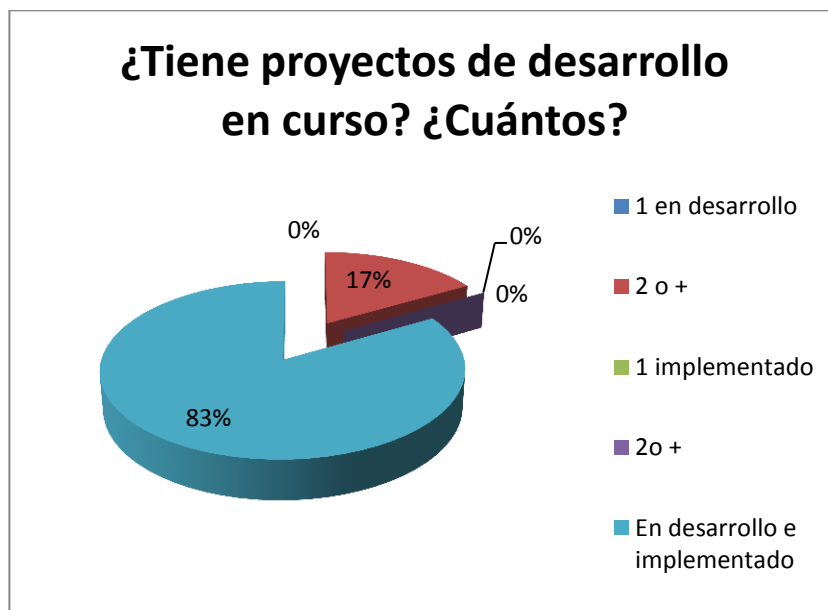
- 1) Gestión back office (ERP)
- 2) Gestión front office (CRM, SRM)
- 3) Inteligencia de negocios
- 4) A la medida
- 5) Facturación Electrónica
- 6) Consultoría
- 7) Capacitación
- 8) Bancarios
- 9) Móvil
- 10) Licenciamiento de aplicativos utilitarios
- 11) Otros _____



Entre la gama de opciones de la cartera de productos de las empresas, las opciones de inteligencia de negocios, facturación electrónica, servicios de consultoría tienen el 14% cada una, del total de las empresas consultadas. Le sigue la opción de ERP, desarrollo de software a la medida, capacitación y aplicativos móviles tienen, cada una, el 11% del total. Con el 7% le sigue la opción CRM, y con el 4%, cada una, las opciones de aplicativos móviles y licenciamiento de aplicativos utilitarios.

7. ¿Tiene proyectos de desarrollo en curso? ¿Cuántos?

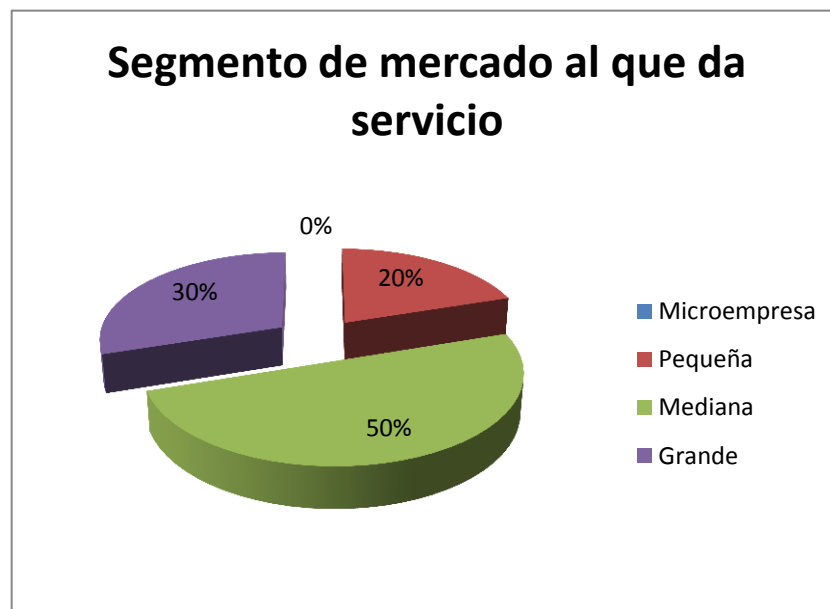
- 1) 1 proyecto en desarrollo
- 2) 2 o más proyectos en desarrollo
- 3) 1 proyecto implementado
- 4) 2 o más proyectos implementados
- 5) Proyectos en desarrollo e implementados



En esta pregunta, el 83% de las empresas entrevistadas tienen proyectos en desarrollo e implementados, frente a un 17% que señaló 2 o más proyectos en desarrollo.

8. Segmento de mercado al que da servicio

- 1) Microempresa
- 2) Pequeña
- 3) Mediana
- 4) Grande

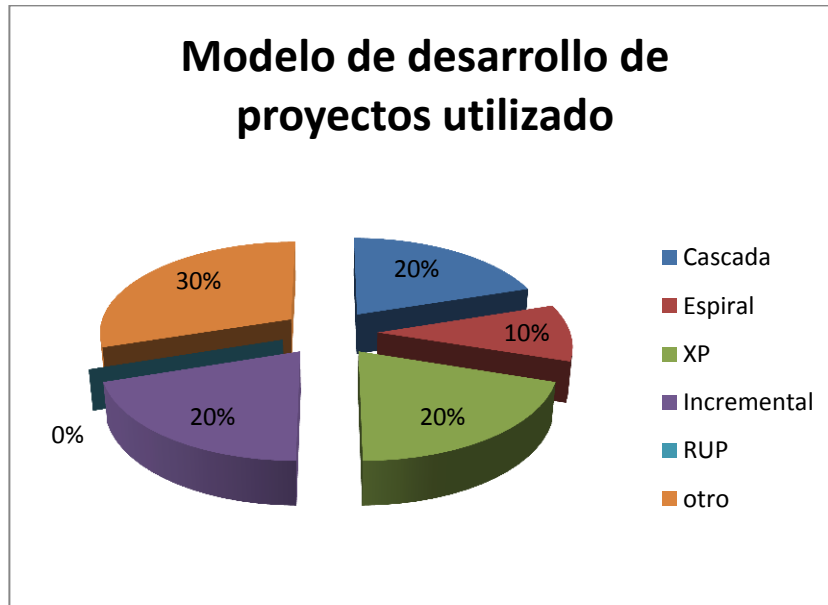


De todas las empresas consultadas, el 50% de ellas manifestaron dar servicio a la mediana empresa, en tanto que el 30% ofrece servicios a las grandes empresas y el 20% a la pequeña empresa.

PARTE TÉCNICA

9. Modelos de desarrollo de proyectos utilizados:

- 1) cascada
- 2) espiral
- 3) programación extrema
- 4) incremental
- 5) rup
- 6) otro

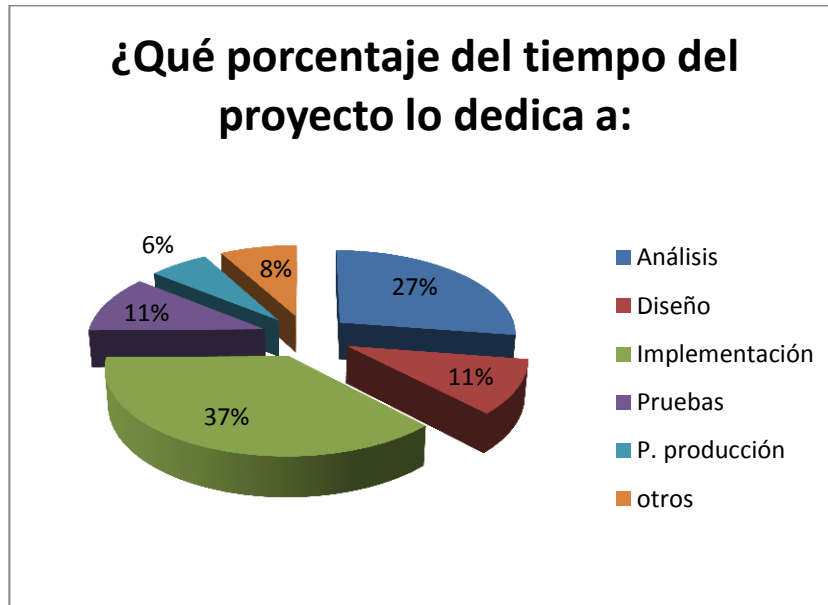


En cuanto a los modelos de proyectos de desarrollo de software utilizados, el 30% de las empresas consultadas se pronunciaron por la opción “otro”; en partes iguales (20%) se pronunciaron por los modelos Incremental, Programación Extrema y Cascada, en tanto que un 10% se inclinan por el modelo Espiral.

10. ¿Qué porcentaje del tiempo del proyecto lo dedica a

- 1) análisis
- 2) diseño
- 3) implementación
- 4) pruebas
- 5) puesta en producción

Justifique la respuesta _____



El 37% de los entrevistados manifestó dedicar mayor parte de su tiempo en la implementación; el 27% de las empresas entrevistadas dedican mayor parte de su tiempo al análisis. En un porcentaje igual (11%), se encuentran las opciones Diseño y Pruebas, seguido de las opciones Otros con el 8% y Puesta en producción con el 6%

11. ¿Qué porcentaje del costo del proyecto lo destina a

- 1) Gestor de proyecto
- 2) Analistas
- 3) Desarrolladores
- 4) Equipo de pruebas

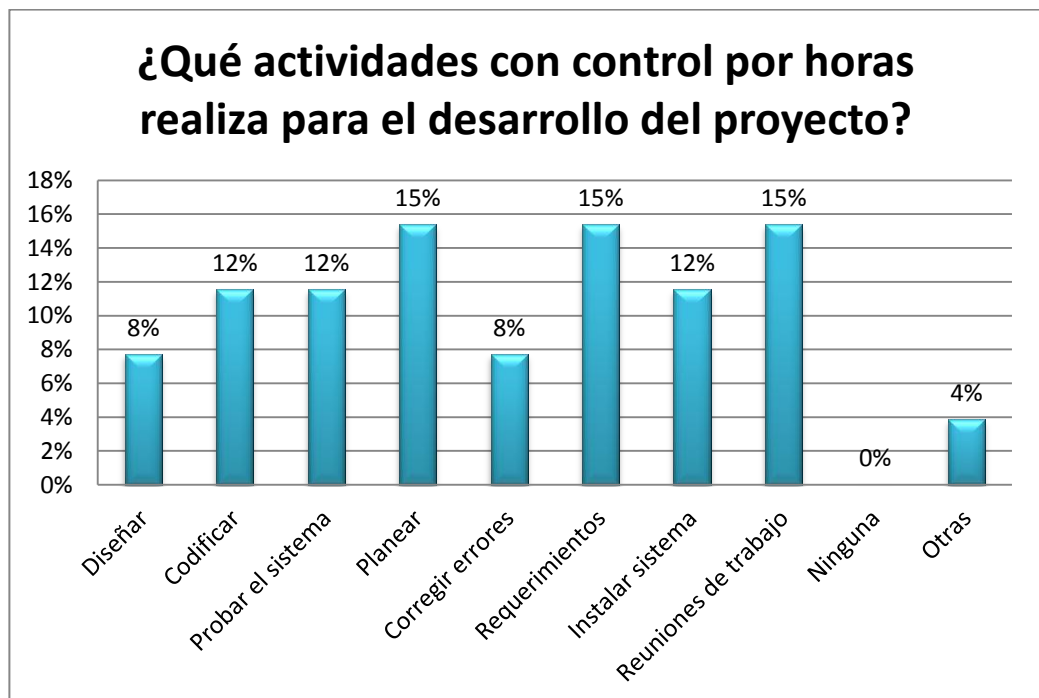
La respuesta a esta pregunta por parte de uno de los entrevistados no fue proporcionada, por cuanto consideró que se estaba indagando sobre información delicada de la empresa, por lo tanto, no se pudo realizar la tabulación de los datos.

12. ¿Cómo definiría métrica de software?

Al definir métrica de software, 2 de los entrevistados respondieron que este término tiene que ver con aspectos relacionados con la calidad del software; otros 2 entrevistados señalaron que métrica de software significa rendimiento del software. Por último, 1 entrevistado señaló que el término antes mencionado significa los pasos a seguir para el desarrollo de un proyecto de software.

13. ¿Qué actividades con control por horas realiza para el desarrollo del proyecto?

- 1) diseñar
- 2) codificar
- 3) probar el sistema
- 4) planear
- 5) corregir errores
- 6) requerimientos
- 7) instalar el sistema
- 8) reuniones de trabajo

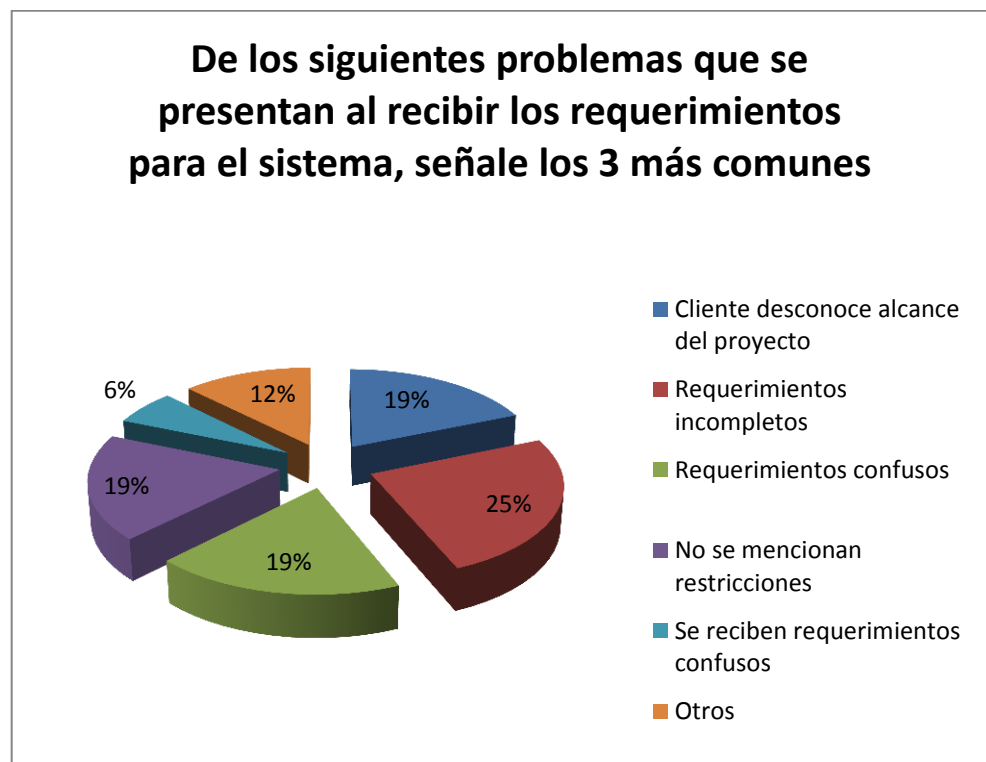


Con el 15% cada una se encuentran las opciones Planear, Requerimientos y Reuniones de trabajo; seguido de Codificar, Probar el sistema e Instalar el sistema. Con el 8% cada una, las opciones de Diseñar y Corregir errores. Finalmente con el 4% la opción Otras, como los porcentajes que se definieron en cuanto se relaciona a la actividad de control por horas.

14. De los siguientes problemas que se presentan al recibir los requerimientos para el sistema, señale los 3 más comunes

- 1) cliente desconoce el alcance del proyecto
- 2) se reciben requerimientos incompletos
- 3) se reciben requerimientos confusos
- 4) no se mencionan restricciones
- 5) se reciben requerimientos agrupados
- 6) otros

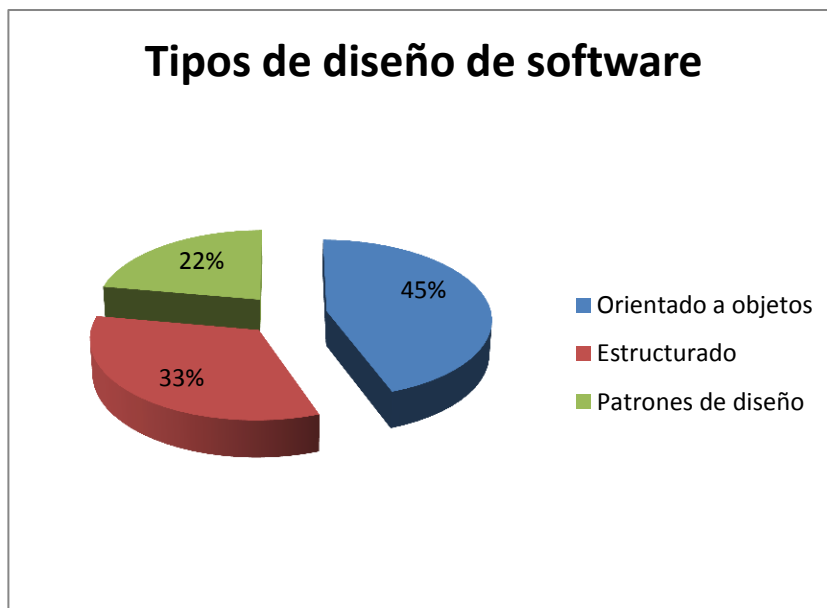
¿Cuáles? _____



El 25% de los consultados señaló que uno de los problemas principales para el desarrollo del sistema los requerimientos incompletos, seguido en partes iguales (19%) las opciones Cliente desconoce alcance del proyecto, Requerimientos confusos y No se mencionan restricciones. En tanto que el 12% señala Otros y el 6% Se reciben requerimientos confusos.

15. Tipos de diseño del software

- 1) orientado a objetos
- 2) estructurado
- 3) patrones de diseño
- 4) otros



El 45% de las empresas entrevistadas se pronunció por la opción Orientado a objetos el tipo de diseño de software utilizado, seguido por el 33% de los entrevistados con el tipo de diseño Estructurado, y finalmente el 22% con Patrones de diseño.

16. ¿Hace inspección en la fase de diseño?

- 1) si
- 2) no

La respuesta a esta pregunta señala que el 80% de las empresas entrevistadas si hace inspección en la fase de diseño, mientras que el 20% señala que no.



17. ¿Hace reutilización de software?

- 1) Si
- 2) no
- 3) desconoce

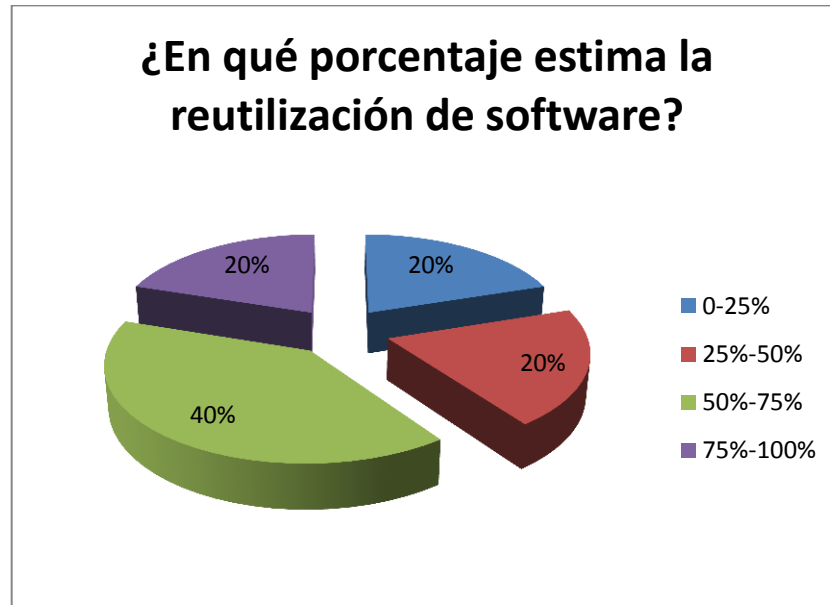
¿Por qué? _____



El 100% de las empresas entrevistadas señala que si hace reutilización de software.

18. ¿En qué porcentaje estima la reutilización?

- 0-25%
- 25%-50%
- 50%-75%
- 75%-100%

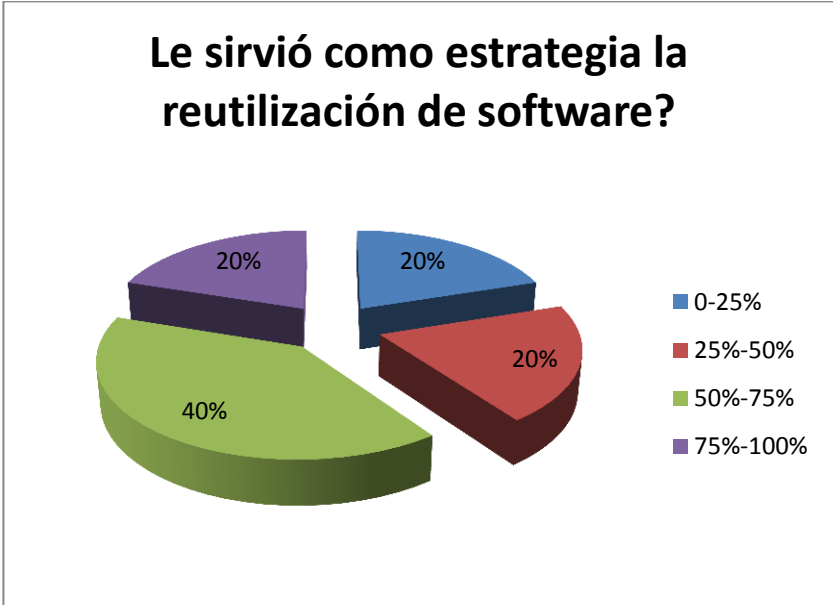


En cuanto al porcentaje de reutilización, el 40% de los entrevistados reutiliza en el rango del 50%-75%; en partes iguales (20%) las opciones 0%-25%, 25%-50% y 75%-100%

19. ¿Le sirvió como estrategia? Justifique la respuesta

- 1) 0-25%
- 2) 25%-50%
- 3) 50%-75%
- 4) 75%-100%

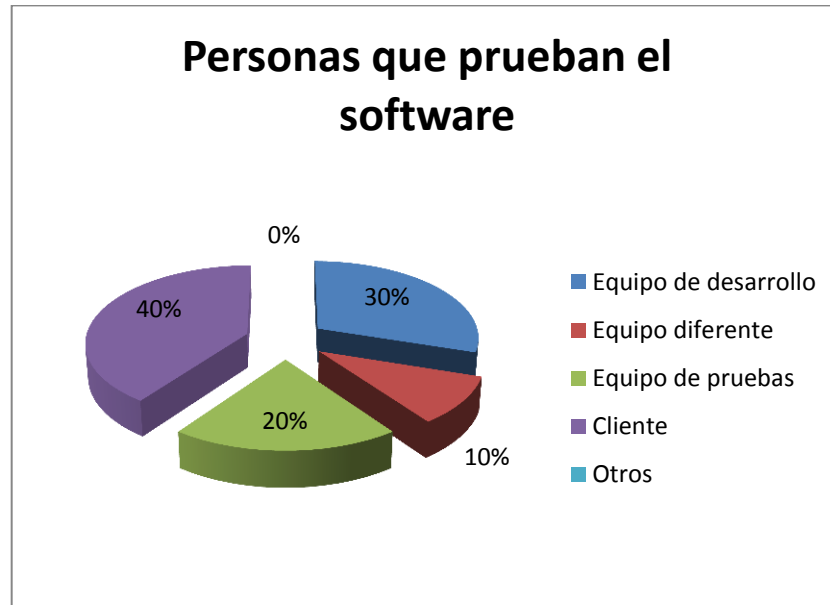
En cuanto se refiere a esta pregunta, el 40% de las empresas consultadas, señaló que la reutilización la estiman entre el 50%-75%, en tanto que en partes iguales (20%), la estrategia de reutilización de software sirvió 0%-25%, 25%-50% y 75%-100%



20. Personas que prueban el software

- 1) el equipo de desarrollo
- 2) un equipo de desarrollo diferente
- 3) un equipo de pruebas
- 4) el cliente
- 5) otros. Especifique_____

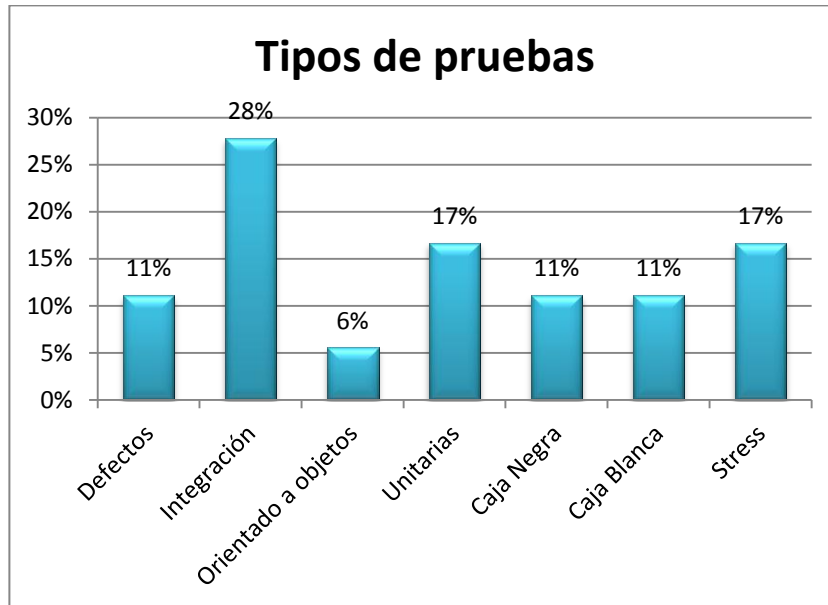
Del total de los entrevistados, el 40% señaló como personas que prueban el software el Cliente, seguido del 30% con el Equipo de desarrollo, el 20% el Equipo de pruebas y el 10% con un Equipo diferente.



21. Tipos de pruebas

- 1) defectos
- 2) integración
- 3) orientado a objetos
- 4) unitarias
- 5) caja negra
- 6) caja blanca
- 7) stress

En cuanto al tipo de pruebas, el 28% de las empresas entrevistadas señaló que realizan las pruebas de Integración al software. Repartido en la misma proporción (17%), se encuentran las pruebas Unitarias y Stress; compartiendo el mismo porcentaje (11%) se ubican las pruebas de Defectos, Caja negra y Caja Blanca y, por último las pruebas Orientadas a objetos con el 5% de las preferencias.

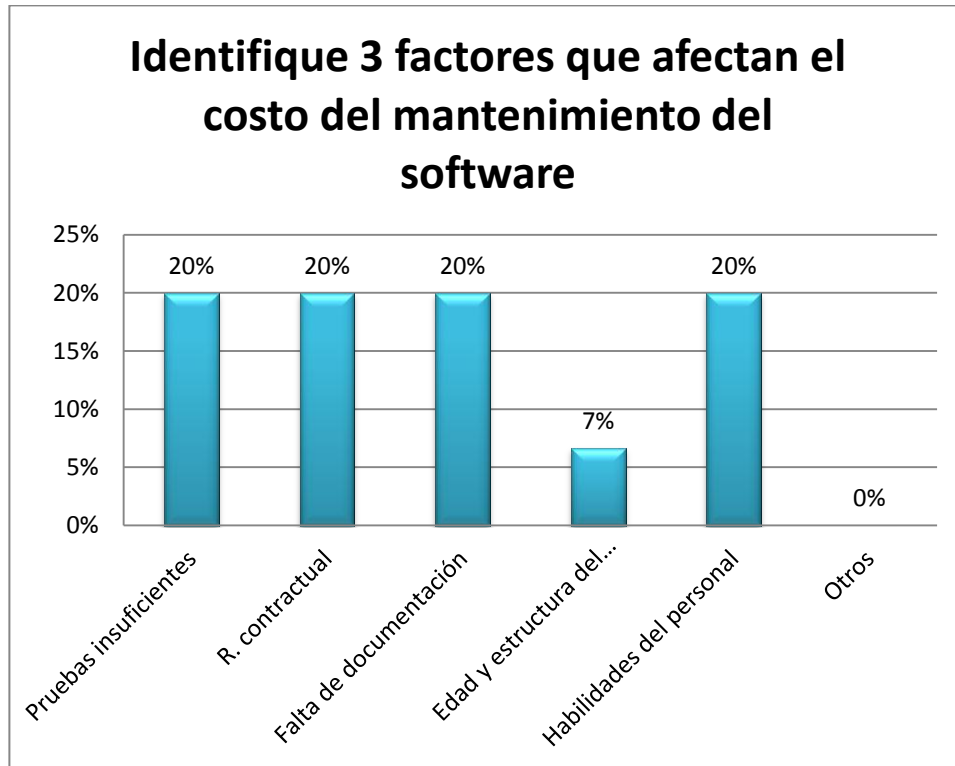


22. Identifique 3 factores que afectan el costo del mantenimiento del software.

- 1) rotación de personal
- 2) pruebas insuficientes
- 3) responsabilidad contractual
- 4) falta de documentación
- 5) edad y estructura del programa
- 6) habilidades del personal
- 7) otros

¿Cuáles? _____

Del total de los entrevistados y con un porcentaje igual (20%), se encuentran las opciones Pruebas insuficientes, Responsabilidad contractual, Falta de documentación y Habilidades del personal, como factores que afectan el costo del mantenimiento del software.

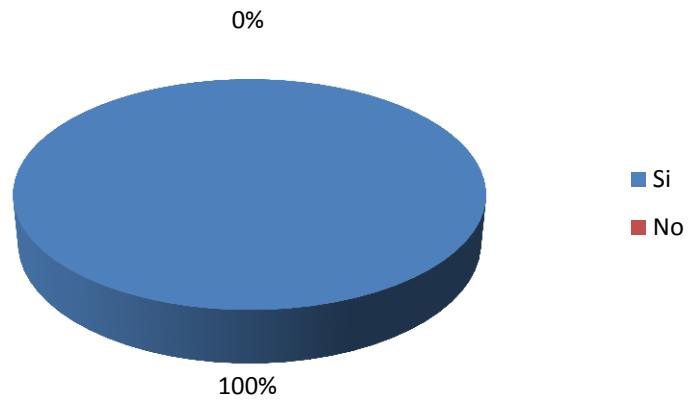


23. ¿Corrige los defectos del software al ser reportados?

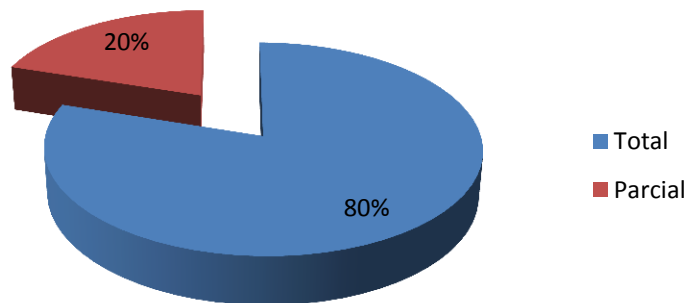
- | | | |
|-------|-------|---------|
| 1) Si | Total | Parcial |
| 2) no | | |

El 100% de las empresas entrevistadas manifestó corregir los defectos del software al ser reportados. Asimismo, el 80% de aquellas corrige totalmente los defectos y el 20% restante lo hace parcialmente.

¿Corrige los defectos del software al ser reportados?



Corrección de defectos



24. ¿Registra el tiempo que utiliza en corregir los defectos?

- 1) si
- 2) no



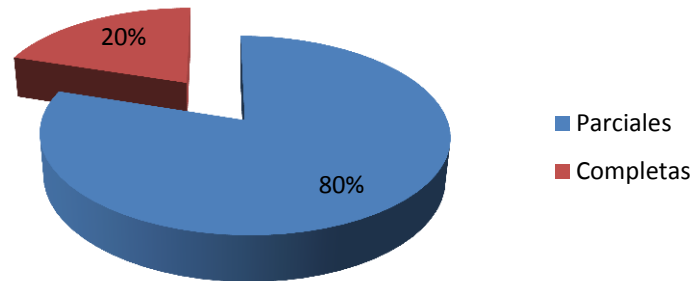
De todas las empresas entrevistadas, el 60% de ellas si registra el tiempo que utiliza en corregir los defectos, en tanto el 40% restante no los registra.

25. Para corregir el mal funcionamiento del software realiza

- 1) reinstalaciones parciales
- 2) reinstalaciones completas

La respuesta a esta pregunta fue: el 80% realiza reinstalaciones parciales y 20% reinstalaciones completas.

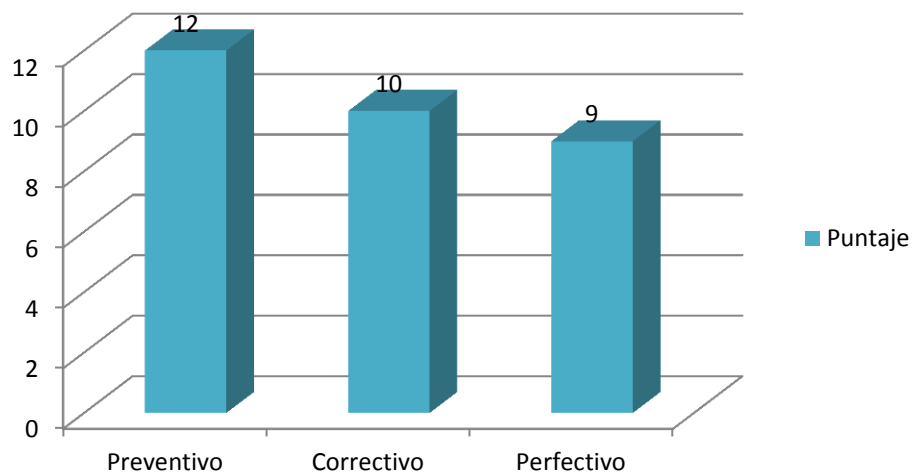
Para corregir el mal funcionamiento del software realiza reinstalaciones:



26. Ordene por prioridad: El tipo de mantenimiento es

- 1) perfectivo (mejora la funcionalidad del software)
- 2) correctivo (repara las fallas en el software)
- 3) preventivo

Tipo de mantenimiento



En orden de importancia, las empresas señalaron el mantenimiento Preventivo como el que se encuentra en primer lugar para su aplicación, seguido del correctivo y, por último el perfectivo.

4.5. Entrevista para medir la satisfacción del usuario

A la par de la realización de la entrevista para conocer datos sobre las empresas de desarrollo de software, se piensa realizar otra entrevista para responder interrogantes que permitan medir la satisfacción y las preferencias del usuario.

Así como en la entrevista final a las empresas de desarrollo de software, la entrevista para medir la satisfacción del usuario está tomada de la misma muestra intencional que proporciona los datos para la entrevista inicial.

4.6. Análisis de resultados

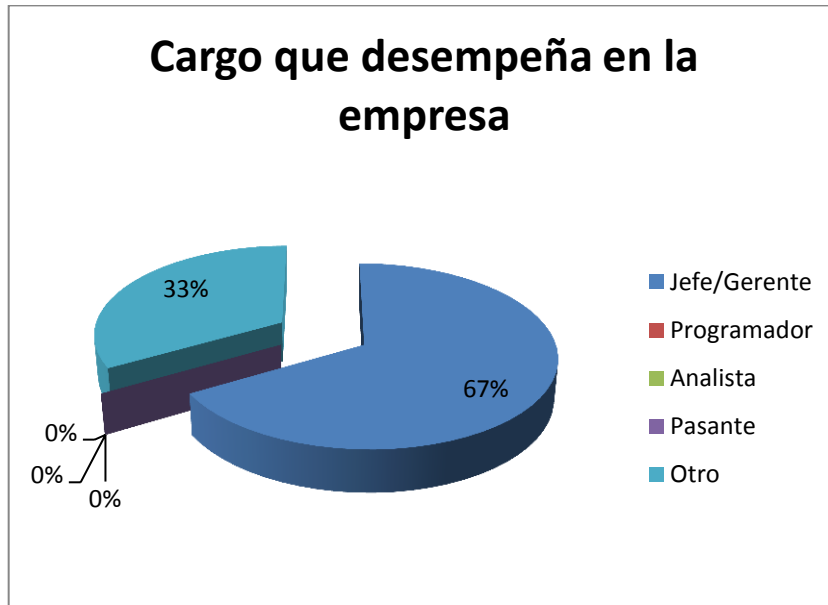
ENTREVISTA (2) A EMPRESAS DE DESARROLLO DE SOFTWARE

FECHA: _____

OBJETIVO: Conocer el nivel de aceptación de los usuarios con relación al servicio prestado.

DATOS BÁSICOS DE LA EMPRESA:

- 1. Nombre de la empresa.**
- 2. Cargo que desempeña en la empresa**



De las empresas visitadas, el 67% de ellas señala que la persona entrevistada es Jefe/Gerente de proyectos, mientras que el 33% tiene otro tipo de cargo.

DATOS DE PRESTACIÓN DE SERVICIO DE SOPORTE

3. ¿Tiene un departamento o área para soporte a usuario?

Si

No

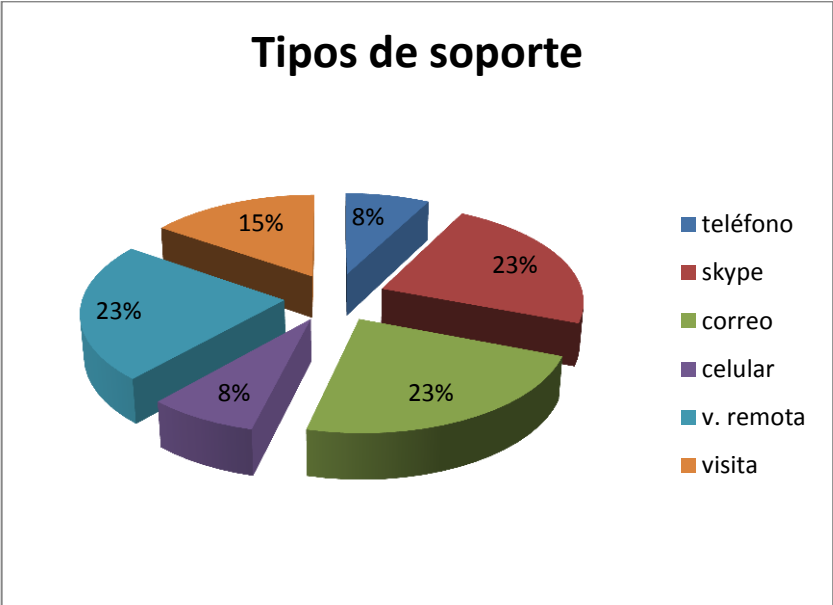
El 100% de los encuestados manifestó si tener implementado un departamento de soporte a usuario.



4. ¿Cómo brinda el soporte a usuario? Señale al menos 3 opciones

- Vía telefónica
- Vía Skype
- Vía correo electrónico
- Vía celular
- Vía escritorio remoto
- Visita personal del técnico-analista

Como tipo de soporte se señala Skype, correo electrónico y vía escritorio remoto, con el 23% cada opción, como las más utilizadas para brindar servicio a los usuarios. Le sigue Visita personal del técnico-analista, con 15%, y por último las opciones celular y teléfono con el 8% cada una, como las señaladas por los entrevistados.



5. ¿Registra los requerimientos de los usuarios?

Si

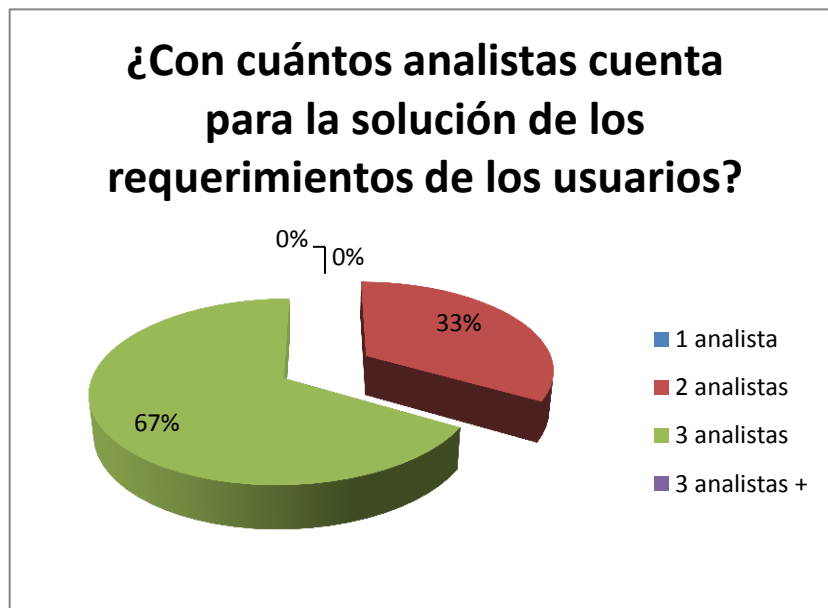
No



El 100% de los entrevistados señaló que si registra los requerimientos de los usuarios cuando son reportados.

6. ¿Con cuántos analistas cuenta para la solución de los requerimientos de los usuarios?

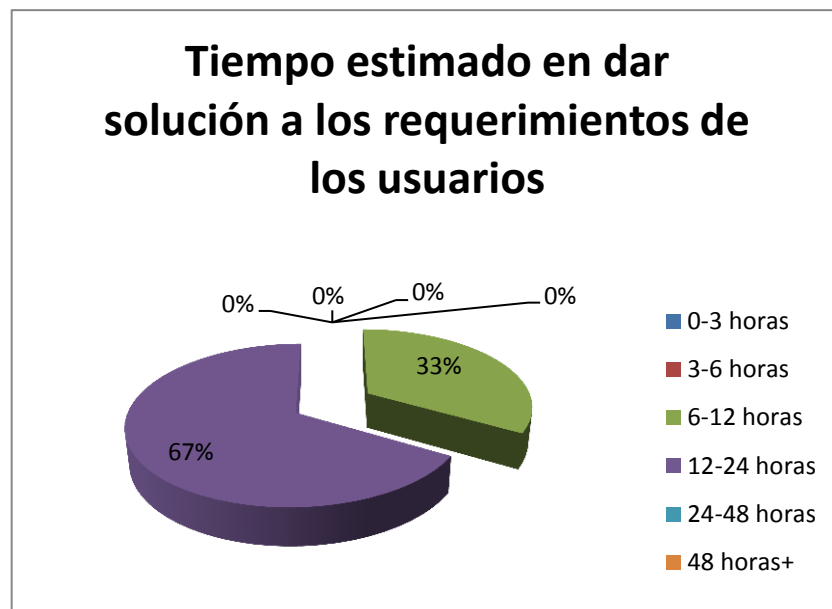
- 1 analista
- 2 analistas
- 3 analistas
- Más de 3 ¿Cuántos?_____



El 67% de todas las empresas entrevistadas señala que dispone de 3 analistas para la resolución de los requerimientos de los usuarios, en tanto que el 33% señala que dispone de 2 analistas para tal cometido.

7. Señale el tiempo estimado en que se da solución a los requerimientos de los usuarios

- De 0 a 3 horas
- Más de 3 horas hasta 6 horas
- Más de 6 hasta 12 horas
- Más de 12 horas hasta 24 horas
- Más de 24 horas hasta 48 horas
- Más de 48 horas. ¿Cuántas? _____

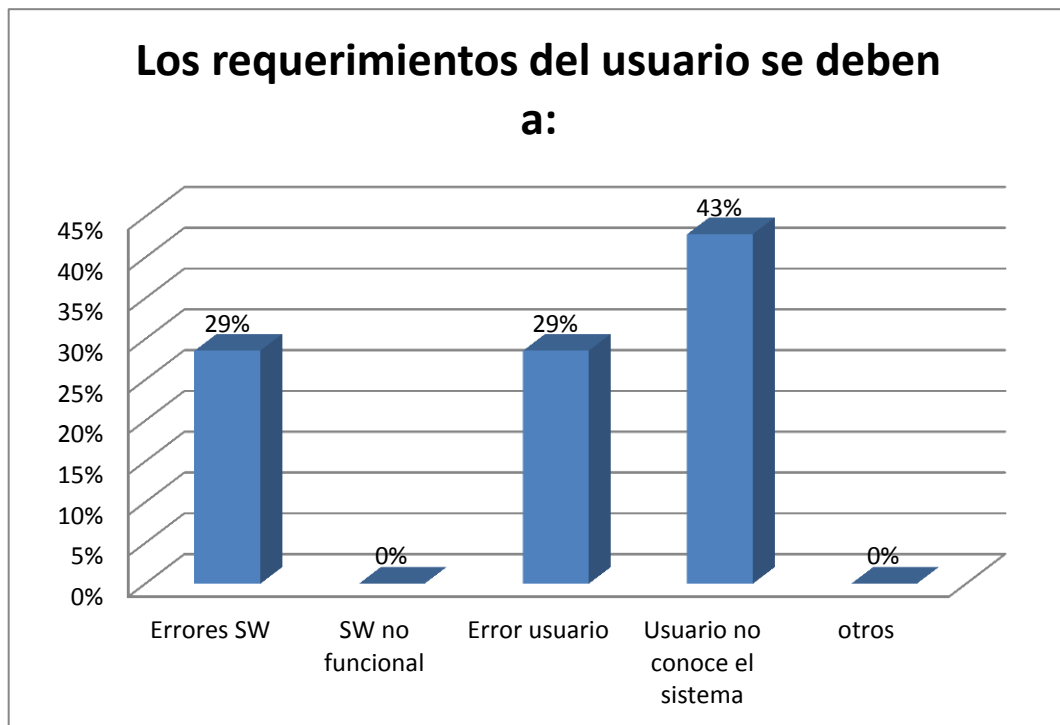


En respuesta a esta pregunta, el 67% de las empresas entrevistadas manifestó que el tiempo que ellas estiman para solucionar los requerimientos de los usuarios lo brindan entre 12-24 horas. El 33%, por su parte señala que la solución a los requerimientos la brindan entre 6-12 horas.

8. Los requerimientos del usuario se deben a:

- Errores en el software
- El software no es funcional
- Errores de usuario
- Usuario no conoce el sistema
- Otros. ¿Cuáles? _____

De acuerdo a las empresas entrevistadas, el 43% señala como más frecuente en relación con la petición de solución de requerimientos, a que el Usuario no conoce el sistema, seguido de las opciones de Errores de software y Errores de usuario, con 29% cada una de ellas.

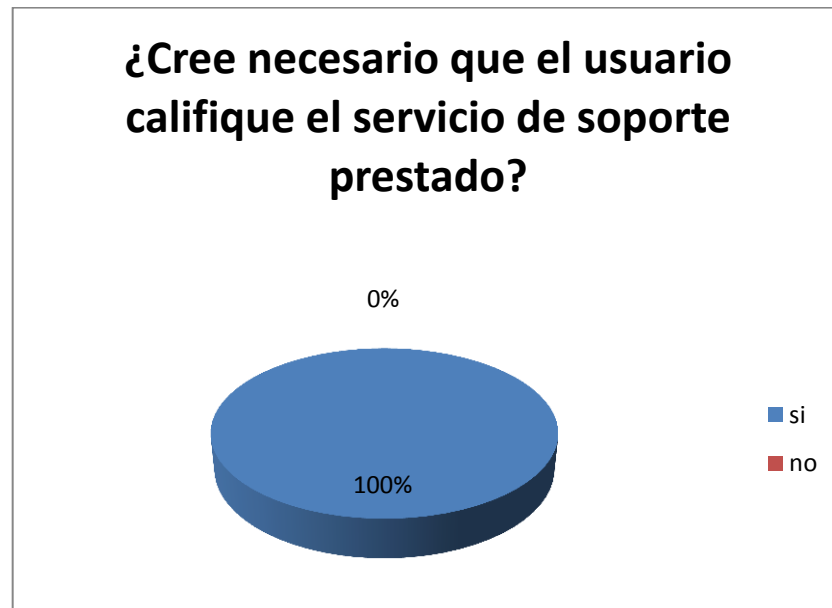


MEDICIÓN DE LA SATISFACCIÓN DEL USUARIO

9. ¿Cree necesario que el usuario califique el servicio de soporte prestado?

Si

No



En lo relacionado con la parte de la entrevista que busca medir la satisfacción del usuario, la pregunta 9 tuvo el 100% de las empresas que respondieron que consideran que el usuario si debe calificar el servicio de soporte brindado.

10. ¿El usuario califica el servicio prestado por el técnico-analista al solucionar el requerimiento?

Si

No

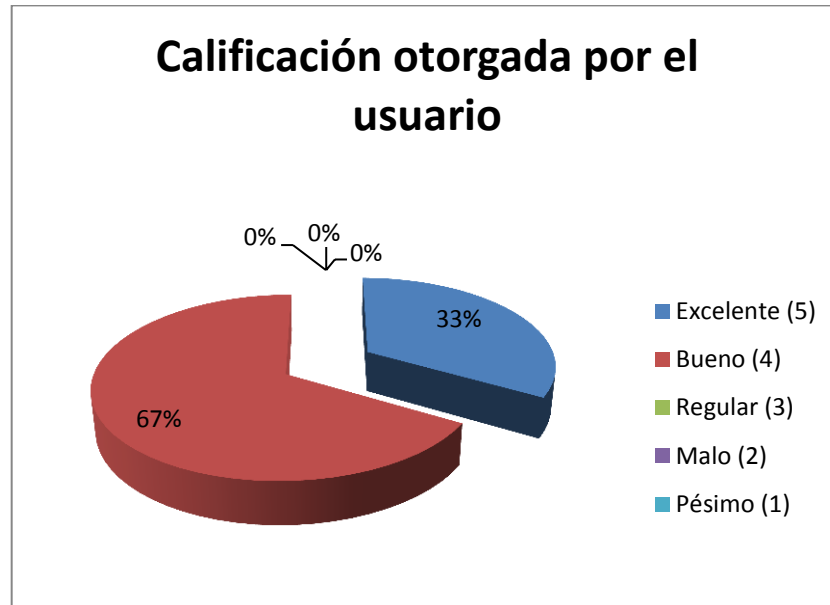
El 100% de los entrevistados señaló que el usuarios si califica el servicio recibido del técnico-analista.



11. Si es afirmativa la respuesta anterior y de acuerdo a su perspectiva, ¿cuál es la calificación otorgada por el usuario? Califíquese de 1 a 5

- Excelente (5)
- Bueno (4)
- Regular (3)
- Malo (2)
- Pésimo (1)

La respuesta de las empresas entrevistadas en relación con esta pregunta, dejó de manifiesto que la calificación de los usuarios al servicio prestado por el departamento o área de soporte es Bueno (67%), mientras que la opción de Excelente se ubicó en segundo lugar (33%).

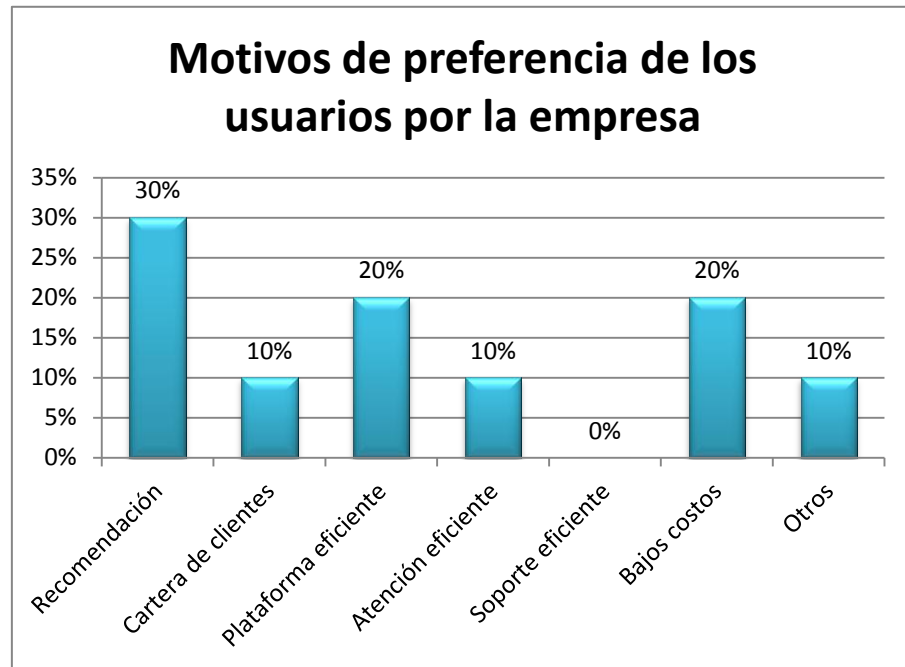


12. ¿Cuáles cree usted que fueron los motivos, en general, que ha hecho que sus clientes prefieran su empresa antes que a otras que ofrecen el mismo servicio? Señale por lo menos 3 motivos

- Por recomendación de otras personas
- Por la cartera de clientes atendidos
- Por la eficiencia de la plataforma adquirida
- Por la eficiencia de la atención al cliente
- Por la eficiencia en el soporte a usuario
- Por los bajos costos en los productos ofertados
- Otros _____

De las empresas entrevistadas, el 30% de ellas señaló que los usuarios han preferido sus servicios al de otras empresas Por recomendación de otras personas; en igual proporción (20% cada una) se encuentran las opciones Por la eficiencia de la plataforma adquirida y Por los bajos costos en los productos ofertados. Las opciones Por la cartera de

clientes atendidos, Por la eficiencia en la atención al cliente y Otros, registran un porcentaje del 10% cada una de ellas.



Concluida la tabulación de los datos de las entrevistas aplicadas a las empresas de desarrollo de software y obtenidos los resultados finales, se requiere elaborar el informe final, que es tratado en el capítulo siguiente.

CAPÍTULO V: INFORME FINAL

Como resultado de la tabulación de los datos, tanto de la entrevista para determinar datos generales de las empresas de desarrollo de software asociadas a la AESOFT y otras empresas dedicadas a la misma actividad, en cuanto a metodologías utilizadas y otros aspectos concernientes para la administración de proyectos, como de la entrevista que sirvió de base para medir las preferencias de los usuarios, se obtuvo información de importancia para descubrir, en parte, el funcionamiento de las mismas.

Cabe recalcar que, debido a la poca colaboración y el poco interés prestado por algunas empresas, hubo la necesidad de tomar una muestra intencional de las mismas.

La investigación propiamente dicha inicia con un estudio piloto a 5 empresas. Como resultado de la aplicación de ese piloto, se constató que existían preguntas agrupadas y de difícil tabulación, por lo que para la entrevista definitiva, el banco de preguntas fue corregido y modificadas algunas de ellas; además, no todas las interrogantes preparadas respondían a los objetivos iniciales, por lo que se elaboró un nuevo cuestionario para aplicarlo a las empresas.

Otro de los inconvenientes presentados fue el contacto con los directivos de las empresas. Para subsanar este obstáculo, de la lista emitida por la AESOFT se tuvo que elegir una muestra intencional para realizar la entrevista. Las empresas que permitieron la realización de la misma, tienen constancia del estudio por documentación a ellas presentada, otorgada por la Facultad.

De la entrevista para recoger datos generales de las empresas y datos sobre metodologías utilizadas para el desarrollo de proyectos de software, se puede inferir que las empresas guayaquileñas que desarrollan e implementan software se mantienen en el rango de pequeña empresa con pocos años de

funcionamiento en el mercado local, las que se encuentran dirigidas por el género masculino en su totalidad hasta el momento. Las metodologías utilizadas tanto para el desarrollo como para la gestión de proyectos, se conoce que las empresas utilizan tanto métodos evolutivos como metodologías ágiles, con un ligero incremento de estas últimas, según información de los entrevistados. Los servicios que brindan a los usuarios consisten en una variada cartera de productos, ofertada para tanto para pequeña, como para mediana y grande empresas. Los problemas a los que los desarrolladores se ven enfrentados se relacionan con una mala definición del proyecto: el usuario desconoce el alcance del mismo, lo que motiva a que se planteen mal los requerimientos y, por ende, el desarrollo del proyecto no cubra las expectativas del cliente.

Por otro lado, de la entrevista para medir la satisfacción del usuario, se pudo deducir que para el servicio de soporte a usuario, las empresas de desarrollo cuentan con un departamento para tal cometido, con un determinado número de analistas que atienden los requerimientos de los usuarios. Estos requerimientos, en términos generales, se presentan por el poco conocimiento de los usuarios en relación con el producto de software que se encuentran utilizando. En cuanto a las preferencias de los usuarios, las empresas que desarrollan sistemas son escogidas por recomendación de otras personas y otros factores como los bajos costos en los productos ofertados.

CONCLUSIONES Y RECOMENDACIONES

Al finalizar el trabajo de investigación basado en el estudio sobre a la gestión de proyectos de software realizado a las empresas de desarrollo de software de Guayaquil afiliadas a la AESOFT y otras empresas dedicadas a la misma actividad, se puede afirmar que el cumplimiento de los objetivos que se persiguieron al inicio del mismo, se llevó a cabo íntegramente.

Conclusiones

La entrevista realizada a las empresas permitió determinar el tiempo de actividad en el mercado local de software, tomando en consideración el tipo de empresas de acuerdo al número de empleados que tiene cada una. Es así como se estableció que el mayor porcentaje de las empresas de desarrollo de software de Guayaquil son pequeñas empresas y su tiempo promedio de funcionamiento se encuentra en el rango de 5-8 años.

Los modelos ágiles de desarrollo de software han tenido su despunte en los últimos cinco años. Esto se puede evidenciar en las preferencias de los entrevistados al colocar estas metodologías de desarrollo a nivel de los métodos evolutivos, comúnmente utilizados para el desarrollo e implementación de proyectos.

Para complementar lo antes mencionado, cabe señalarse que los métodos evolutivos son cíclicos, siguen el proceso normal del desarrollo de software y su desarrollo se lo efectúa en un centro de cómputo. Las metodologías ágiles, entre ellas se señalan a la Programación Extrema XP y SCRUM, por el contrario son metodologías que tienen como característica el trabajo con el usuario final del sistema para la retroalimentación de la información. Por tal motivo, los escritorios de los desarrolladores del sistema deben ser colocados al lado del escritorio de los usuarios.

La entrevista para medir la satisfacción del usuario con relación al servicio prestado por las empresas que desarrollan software, permitió conocer cuán importante es para los usuarios recibir soporte una vez implementado el sistema. Las empresas que otorgan servicio consideran necesaria la calificación del usuario en cuanto a su servicio se refiere; por tanto el nivel de aceptación del usuario al servicio que otorgan las empresas se encuentra preferentemente en el rango de bueno con una tendencia a excelente, de acuerdo a la calificación que se le otorgó a cada una de las opciones.

Recomendaciones

En relación con la verificación de la efectividad del software, se pueden anotar algunas recomendaciones que podrían ser adoptadas por las empresas que desarrollan software y los usuarios de los sistemas desarrollados.

Los distintos sistemas que son implementados en las empresas, como se conoce, son analizados y diseñados por el grupo de desarrollo que tiene a su cargo ese cometido; el gestor de proyectos es el encargado de recabar y entender los requerimientos de los usuarios para trasladarlos a su equipo, en donde se definirá el plan de trabajo a seguir, que incluye el tipo de metodología de desarrollo. Aunque la elección de la metodología dependerá de la complejidad del sistema a desarrollar, la utilización de las metodologías que permitan mayor retroalimentación del usuario hacia el nuevo sistema y que aquel sea parte integrante en su construcción, sería adecuado para el éxito del proyecto; entre estas metodologías se encuentran las ágiles como Programación Extrema o SCRUM. Hay que recordar que las metodologías ágiles también se enmarcan dentro del simple diseño, por lo que elaborar un sistema de fácil comprensión para el usuario sería uno de los objetivos de la empresa que desarrolla software.

Junto con el uso y aplicación de las metodologías de desarrollo ágiles, es indispensable la certificación de los profesionales que trabajan en la construcción de sistemas, ya que una empresa que cuenta con este capital humano de profesionales capacitados, puede ubicarse dentro de un sitio preferencial dentro del competitivo mercado de las empresas que desarrollan software.

Para la realización de las pruebas del software, sería necesario que, como nueva política a implantarse, se seleccione y designe un equipo de pruebas, porque éste estaría únicamente encargado de ese cometido y podría hacer un examen exhaustivo del producto antes de que lo pruebe el cliente.

Por último, para evitar el incremento en el costo del mantenimiento del software, sería conveniente tener en cuenta algunos aspectos que podrían reducir este inconveniente o que sea inexistente. Como primer punto, mantener una estabilidad laboral para los empleados, ya que por ascensos o por la búsqueda de mejores sueldos y beneficios sociales, hace que los trabajadores abandonen su lugar de trabajo y quede el sistema por ellos desarrollado, en manos de gente nueva que desconoce los pormenores del mismo. Del mismo modo, la elección de capital humano con sólidos conocimientos que pase a formar parte del equipo de trabajo, garantizará la eficiencia en la calidad del producto desarrollado.

REFERENCIAS BIBLIOGRÁFICAS

- AESOFT. (2011). *Asociación Ecuatoriana de Software*. Recuperado el 2013, de <http://aesoft.com.ec/www/index.php/quienes-somos>
- Aparicio, A. (Diciembre de 2012). *eBibiloUnad*. Recuperado el 2013, de <http://datateca.unad.edu.co/contenidos/301404/301404.pdf>
- Arias, F. G. (2006). *El proyecto de investigación* (Quinta Edición ed.). Caracas, Venezuela: Editorial Episteme C.A. .
- berrie.dds.nl*. (s.f.). Recuperado el 2014, de <http://www.berrie.dds.nl/calcss.htm>
- Cabrera, A., Solano, R., & Montalván , M. (2009). *Procesos de ingeniería de software*. Recuperado el 2013, de Universidad Técnica Particular de Loja:
http://scholar.googleusercontent.com/scholar?q=cache:Z2uxCj7wQZkJ:scholar.google.com/+%22modelo+cascada%22&hl=es&as_sdt=0,5
- Canós, J., Letelier, P., & Penadés, M. C. (2003). *Universidad Politécnica de Valencia*. Recuperado el 2013, de http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf
- Díaz Polo, D., & Delgado Dapena, M. D. (2011). *Definición de un proceso de desarrollo de software en un entorno universitario*. Cuba: D - Instituto Superior Politécnico José Antonio Echeverría. CUJAE.
- Grompone, J. (1996). *Gestión de Proyectos de Software*. Montevideo, Uruguay: La Flor del Itapebí.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2010). *Metodología de la Investigación* (Quinta edición ed.). (J. Mares Chacón, Ed.) Mexico, México DF: McGraw-Hill/Interamericana Editores, S.A de C.V.
- IBM DeveloperWorks. (2010). *Rational Team Concert para proyectos SCRUM*. Recuperado el 2013, de <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Rational+Team+Concert+for+Scrum+Projects/page/SCRUM+como+metodolog%C3%ADa>

- Marco Galindo, M. J., & Marco Simó, J. M. (2010). *Escaneando la informática*. España: Editorial UOC.
- Pressman, R. (2005). *Ingeniería del Software* (Sexta ed.). México D.F., México: McGraw-Hill/Interamericana Editores, S.A de C.V.
- Rivera Martínez, F., & Hernández Chávez, G. (2010). *Administración de proyectos. Guía para el aprendizaje* (Primera Edición ed.). (L. M. Cruz Castillo, Ed.) Naucalpan de Juárez, México, México: Pearson Educación de México, S.A. de C.V.
- Salazar, D., Villavicencio, M., Macías, M., & Snoeck, M. (2004). *Estudio estadístico exploratorio de las empresas desarrolladoras de software asentadas en Guayaquil, Quito y Cuenca*. ESPOL Proyecto Vllir, Componente 8.
- Secretaria Nacional de Ciencia y Tecnología -SENESCYT-. (2012). *Las nuevas tendencias en la investigación*. Recuperado el 2013, de lkninkjbjbj ljn kj: <http://www.senescyt.gob.ec>
- Sommerville, I. (2004). *Ingeniería del Software* (Séptima Edición ed.). Madrid: Pearson Educación S.A.
- Universidad Autónoma de Sinaloa. (s.f.). *Dirección de Informática*. Recuperado el 2013, de <http://www.uas.edu.mx/galerias/archivos/1376935244.pdf>
- Zavala Ruiz, J. M. (2004). Recuperado el 2013, de <http://claroline.ucaribe.edu.mx/claroline/claroline/backends/download.php?url=L3Bvci1xdWUtZmFsbGFuLWxvcy1wcm95LWRILXNvZnQucGRm&idReset=true&cidReq=NI0215>

GLOSARIO

Buenas Prácticas: Conjunto de habilidades, herramientas y técnicas que pueden aumentar la posibilidad de éxito en una amplia variedad de proyectos.

Datos Cualitativos: Se refiere a descripciones en detalle de situaciones, eventos, personas, interacciones, conductas observadas y sus manifestaciones.

Estructuras de Datos: Es una colección de datos (normalmente de tipo simple) que se caracterizan por su organización y las operaciones que se definen en ellos. Vendrá caracterizada tanto por unas ciertas relaciones entre los datos que la constituyen como por las operaciones posibles en ellas.

Hito: Punto de referencia que señala un evento de importancia en el desarrollo de un proyecto. Comúnmente utilizado para vigilar los progresos de dicho proyecto.

PMI: Project Management Institute. Instituto de Gerenciamiento en Proyectos. El principal objetivo del PMI es diseminar las buenas prácticas en la gestión del proyecto en ocho áreas del conocimiento: alcance, costo, calidad, RH, comunicación, riesgo, suministro y contratación.

Proceso Unificado de Desarrollo de Software: Intento encaminado a reunir los mejores rasgos y características de modelos de procesos de software, pero los caracteriza de manera que implementa muchos de los mejores principios del desarrollo ágil de software.

Subjetivamente: Profundización de experiencias, perspectivas, opiniones, significados del problema en cuestión.

UML: Unified Modeling Language. Es un lenguaje de modelado de sistemas de software, un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

ANEXOS

Anexo 1: Código de Ética de la AESOFT

ARTICULO PRIMERO: Alcance

- 1.1 Este código de conducta profesional para los socios de la Asociación Ecuatoriana de Software (AESOFT) tiene por finalidad establecer y mantener una reputación elevada de integridad y leal competencia profesional entre sus asociados.
- 1.2 La adhesión a las disposiciones de este código es condición indispensable para pertenecer a la Asociación Ecuatoriana de Software.

ARTICULO SEGUNDO: Deberes frente a la Asociación

- 2.1 La Asociación Ecuatoriana de Software persigue los siguientes objetivos (AESOFT, 2011):
 - 2.1.1 Defender los intereses de las empresas asociadas.
 - 2.1.2 Impulsar el desarrollo y perfeccionamiento de las técnicas, métodos y actividades vinculadas con el estudio, diseño, realización, implantación y comercialización de sistemas informáticos que utilicen computadores electrónicos y tareas afines que en la Ciencia Informática se engloban bajo la denominación de Software.
 - 2.1.3 Representar, agrupar, coordinar, organizar y difundir las actividades de las empresas asociadas ante los Poderes públicos, organismos oficiales y privados tanto en el ámbito nacional e internacional.

- 2.1.4 Defender activamente los derechos de propiedad sobre el software en todas sus formas, y orígenes y combatir las prácticas de producción y venta de copias ilegítimas de software.
- 2.1.5 Formular las pautas y promover una ética profesional y un desempeño técnico adecuado para el ejercicio de la actividad.
- 2.1.6 Proceder a intercambiar experiencias, inquietudes e informaciones de toda índole, que tiendan a una mayor eficiencia de las empresas asociadas, como asimismo a desarrollar a la vez un espíritu de cooperación y mutua colaboración entre los mismos.
- 2.1.7 Informar y asesorar sobre la actividad a sus asociados y a toda entidad pública o privada que lo solicite.
- 2.1.8 Organizar y participar en congresos, reuniones, conferencias nacionales e internacionales para difundir y profundizar sus objetivos y otros fines.
- 2.2 Cada asociado debe cooperar a la plena realización de los fines de este código y de los objetivos de la AESOFT.
- 2.3 Cada asociado debe abstenerse de usar software ilegítimo, imponiendo tal deber a sus empleados, cualquiera sea su jerarquía, y adoptando todas las medidas a su alcance para prevenir y evitar los actos y omisiones de sus dependientes que puedan comprometer su responsabilidad.

ARTICULO TERCERO: Deberes frente a otros asociados

- 3.1 Todo asociado debe abstenerse de efectuar comentarios que desacrediten a otros afiliados en sus tratos con el público, en material publicado o con sus clientes y prospectos.

- 3.2 Todo asociado debe abstenerse de formular críticas infundadas o injustificadas a los actos de otros afiliados con la finalidad de atraer para sí sus clientes.
- 3.3 Todo asociado debe actuar con lealtad frente a los demás afiliados, teniendo en consideración los objetivos comunes de los que participan como integrantes de la Asociación Ecuatoriana de Software.

ARTICULO CUARTO: Deberes frente a los clientes

- 1.1 Todo asociado debe trabajar procurando la mejor calidad y la tecnología más adecuada a las necesidades de sus clientes.
- 1.2 Todo asociado debe aconsejar lealmente a sus clientes, atendiendo a sus reales requerimientos e instruyéndolos acerca de la obligatoriedad de usar software legítimo.
- 1.3 Todo asociado debe garantizar la confidencialidad de la información de sus clientes, absteniéndose de usarla en provecho propio o divulgarla a terceras personas.
- 1.4 Todo asociado debe entregar a sus clientes software legítimo, obligándose a no distribuir software con licencias inadecuadas o sin licencia.

ARTICULO QUINTO: Comité de Honor.

- 5.1 El Comité de Honor será integrado por los ex-presidentes de AESOFT y por el comité ejecutivo de AESOFT (Presidente, Vicepresidente y Tesorero) siempre y cuando estos últimos hayan estado por lo menos 3 años consecutivos dentro de la asociación.

ARTICULO SEXTO: Deberes del Comité de Honor.

- 6.1** El Comité de Honor será el encargado de analizar, calificar y aprobar a las personas y empresas que solicitan admisión a la Asociación Ecuatoriana de Software.
- 6.2** El Comité de Honor fijará sus propias normas de procedimiento, las que se ajustarán a las reglas y garantías del debido proceso. Sus acciones serán reservadas a fin de preservar los derechos de quienes puedan encontrarse involucrados en una investigación o denuncia.
- 6.3** El Comité de honor, con carácter de órgano asesor de la Comisión Directiva, tendrá a su cargo velar por la plena vigencia de las previsiones de este código de ética profesional y aconsejar la aplicación de las medidas correspondientes en caso de infracciones, tales como:
 - 6.3.1** La observación consiste en advertir al infractor que debe cumplir determinada norma que ha infringido, exhortándolo a no reincidir.
 - 6.3.2** La censura pública es la divulgación de una declaración acerca de la falta cometida y de la calificación que merece el comité de honor.
 - 6.3.3** La suspensión del socio consiste en la eliminación temporal de los registros de AESOFT
 - 6.3.4** La eliminación del socio consiste en la expulsión de AESOFT.
 - 6.3.5** En todos los casos, las medidas a que refieren las disposiciones precedentes deberán ser proporcionadas a la gravedad de las infracciones cometidas, debiendo respetarse la garantía de defensa, y haciendo aplicación de las disposiciones pertinentes.

6.4 El Comité de Honor podrá dictaminar de acuerdo con la convicción que cada uno de sus integrantes se haya formado sobre la cuestión sometida a su consideración, atendiendo especialmente a las pruebas producidas.

Sus pronunciamientos tendrán un carácter decisorio para la Junta Directiva.

Fuente: AESOFT (2013)

Anexo 2: Listado de empresas proporcionado por la AESOFT

Asociados Pertenecientes a la Ciudad de Guayaquil

| <i>Nombre</i> | <i>Teléfono</i> | <i>Email</i> | <i>Ciudad</i> | <i>Sitio Web</i> |
|--|---|--|-------------------|----------------------|
| BANRED | (02) 2523333 / (04) 2523333 | pnarvaez@banred.fin.ec | Quito, Guayaquil | www.banred.fin.ec |
| IRROUTE SOLUTIONS | (04) 2295330 - 2288679 -02394001 | david.duenas@iroute.com.ec | Guayaquil | www.iroute.com.es |
| TROPIDATOS S.A. | 593 04 2203142 - 593 04 2201177 | avico@tropidatos.com | Guayaquil | www.tropidatos.com |
| BIROBID S.A. | 5103222-2843848 | jquispillo@birobid.com | Guayaquil | www.birobid.com |
| TECH NET C.A. | 042201913 | despinoza@technetca.com | Guayaquil | WWW.TECHNETCA.COM |
| EXODO | 097869995/ 04-6019190. | foordova@exodo.com.ec | Guayaquil | www.exodo.com.ec |
| WINNERCORP S.A. | 4-5000007 4-5000008 | elias.sevilano@winnercorp.info.ec | GUAYAQUIL | www.winnerplanet.net |
| C-ENERGYTEK S.A.-SINERGIA SOLUCIONES DE SOFTWARE | 5114449 | miguel.viejo@sinergias.com | Guayaquil | www.sinergias.com |
| IMAGE TECH | 2523721-2523722-2523709 | leonardo@image.com / cecilia@image.com | Guayaquil | www.image.com |
| MEDIA LOGIC S.A. | 045117256/ 042280143/042280109 | LROMANZ@SIAPRE.COM | GUAYAQUIL | WWW.SIAPRE.COM |
| AGROSOFT S.A. | 04 2881333,04 2610303/ 04 2610129,0998150444 | xcardenas@agrosoft.com.ec | Guayaquil | www.agrosoft.com.ec |
| CIMA IT | +593 4 2280217 | lcastro@gmail.com.ec | Guayaquil | www.cimait.com.ec |
| EIKON S.A. | (593-4) 5024990/ 5024993 Ext. 111 | xriera@eikon.com.ec | Guayaquil | www.eikon.com.ec |
| ALTA TECNOLOGIA EN SOFTWARE S.A. | 04 2289898 - 2289171 - 2287305 | rsanchez@altitecsoft.net | Guayaquil | www.altitecsoft.com |
| SIPECOM S.A. | 2561513 - 2561514 | svinos@specom.com | Guayaquil | www.sipecom.com |
| DESARROLLO INTEGRAL DE SOLUCIONES EMPRESARIALES DIRES S.A. | (593) 4 2598292 / 3731390 EXT 1701 | fausto.ruiz@grupodifare.com | Guayaquil | www.dires.com.ec |
| GEEKS ECUADOR; MODERZACORP S.A. | 5113417 / 091214085 / | adrian.bajana@geeks.com.ec | Guayaquil | http://geeks.com.ec/ |
| TFASE S.A. | 04 2810801/ | kgabor@tfase.com | Duran y Guayaquil | www.tfase.com |

Jueves, 28 de noviembre de 2013

Page 1 of 1

Fuente: AESOFT (2013)

Anexo 3: Formato de entrevista

Entrevista piloto a empresas de desarrollo de software

DATOS INFORMATIVOS:

- 1. Género del entrevistado**
 - 1.1 Hombre
 - 1.2 Mujer
- 2. Cargo que ocupa en la empresa**
 - 2.1 jefe/gerente de proyectos
 - 2.2 programador
 - 2.3 analista
 - 2.4 pasante
 - 2.5 otro

IDENTIFICACIÓN BÁSICA DE LA EMPRESA

- 3. Su empresa es:**
 - 3.1 Microempresa (1 a 9 trabajadores)
 - 3.2 Pequeña (10 a 49 trabajadores)
 - 3.3 Mediana (50 a 99 trabajadores)
- 4. Tiempo promedio de funcionamiento de la empresa (rango)**
 - 4.1 de 0 hasta 2 años
 - 4.2 más de 2 hasta 5 años
 - 4.3 más de 5 hasta 8 años
 - 4.4 más de 8 años
- 5. Servicios que brinda la empresa**
 - 5.1 Implementación de sistemas
 - 5.2 Desarrollo de sistemas
 - 5.3 Ambos
- 6. Tiene proyectos de desarrollo en curso? ¿Cuántos?**
 - 6.1 1 proyecto en desarrollo
 - 6.2 2 o más proyectos en desarrollo
 - 6.3 1 proyecto implementado
 - 6.4 2 o más proyectos implementados
 - 6.5 Proyectos en desarrollo e implementados

PARTE TÉCNICA

- 7. Modelos de desarrollo de proyectos utilizados:**
 - 7.1 cascada

- 7.2 espiral
- 7.3 programación extrema
- 7.4 incremental
- 7.5 rup
- 7.6 otro

8. ¿Qué porcentaje del tiempo del proyecto lo dedica a

- 8.1 análisis
 - 8.2 diseño
 - 8.3 implementación
 - 8.4 pruebas
 - 8.5 puesta en producción
- Justifique la respuesta _____

9. ¿Qué porcentaje del costo del proyecto lo destina a

- 9.1 gestor de proyecto
- 9.2 analistas
- 9.2 desarrolladores
- 9.3 equipo de pruebas

10. ¿Cómo definiría métrica de software?

11. ¿Qué actividades con control por horas realiza para el desarrollo del proyecto?

- 11.1 diseñar
- 11.2 codificar
- 11.3 probar el sistema
- 11.4 planear
- 11.5 corregir errores
- 11.6 requerimientos
- 11.7 instalar el sistema
- 11.8 reuniones de trabajo

12. ¿Qué tipo de problemas se presentan al recibir los requerimientos para el sistema? Ordene por prioridad

- 12.1 cliente desconoce el alcance del proyecto
- 12.2 se reciben requerimientos incompletos
- 12.3 se reciben requerimientos confusos
- 12.4 no se mencionan restricciones
- 12.5 se reciben requerimientos agrupados
- 12.6 otros

¿Cuáles? _____

13. Tipos de diseño del software

- 13.1 orientado a objetos
- 13.2 estructurado
- 13.3 patrones de diseño
- 13.4 otros

14. ¿Hace inspección en la fase de diseño?

- 14.1 si
- 14.2 no

15. ¿Hace reutilización de software?

- 15.1 Si
- 15.2 no
- 15.3 desconoce

¿Por qué? _____
Si la respuesta es afirmativa, en qué porcentaje estima la reutilización?

- 0-25%
- 25%-50%
- 50%-75%
- 75%-100%

¿Le sirvió como estrategia? Justifique la respuesta

- 15.4 0-25%
- 15.5 25%-50%
- 15.6 50%-75%
- 15.7 75%¿100%

16. Personas que prueban el software. Ordene por prioridad

- 16.1 el equipo de desarrollo
- 16.2 un equipo de desarrollo diferente
- 16.3 un equipo de pruebas
- 16.4 el cliente
- 16.5 otros

Especifique _____

17. Tipos de pruebas

- 17.1 defectos
- 17.2 integración
- 17.3 orientado a objetos
- 17.4 unitarias
- 17.5 caja negra
- 17.6 caja blanca
- 17.7 stress

17.8 otro

¿Cuáles? _____

18. Factores que afectan el costo del mantenimiento. Ordene por prioridad

18.1 rotación de personal

18.2 pruebas insuficientes

18.3 responsabilidad contractual

18.4 falta de documentación

18.5 edad y estructura del programa

18.6 habilidades del personal

18.7 otros

¿Cuáles? _____

19. ¿Corrige los defectos del software al ser reportados?

19.1 si

19.2 no

20. ¿Registra el tiempo que utiliza en corregir los defectos?

20.1 si

20.2 no

21. ¿Para corregir el mal funcionamiento del software realiza

21.1 reinstalaciones parciales

21.2 reinstalaciones completas

22. Ordene por prioridad: El tipo de mantenimiento es

22.1 perfectivo (mejora la funcionalidad del software)

22.2 correctivo (repara las fallas en el software)

22.3 preventivo

Elaborado por: la autora

Anexo 4: Formato final de entrevista

Entrevista a empresas de desarrollo de software

FECHA: _____

DATOS INFORMATIVOS:

- 1. Género del entrevistado**
 - 3) Hombre
 - 4) Mujer
- 2. Cargo que ocupa en la empresa**
 - 6) jefe/gerente de proyectos
 - 7) programador
 - 8) analista
 - 9) pasante
 - 10) otro

IDENTIFICACIÓN BÁSICA DE LA EMPRESA

- 3. Su empresa es:**
 - 4) Microempresa (1 a 9 trabajadores)
 - 5) Pequeña (10 a 49 trabajadores)
 - 6) Mediana (50 a 99 trabajadores)
- 4. Tiempo promedio de funcionamiento de la empresa (rango)**
 - 5) de 0 hasta 2 años
 - 6) más de 2 hasta 5 años
 - 7) más de 5 hasta 8 años
 - 8) más de 8 años
- 5. Servicios que brinda la empresa**
 - 4) Implementación de sistemas
 - 5) Desarrollo de sistemas
 - 6) Ambos
- 6. Selecciones las opciones que corresponden a su cartera de productos**
 - 12) Gestión back office (ERP)
 - 13) Gestión front office (CRM, SRM)
 - 14) Inteligencia de negocios
 - 15) A la medida
 - 16) Facturación Electrónica
 - 17) Consultoría

- 18) Capacitación
- 19) Bancarios
- 20) Móvil
- 21) Licenciamiento de aplicativos utilitarios
- 22) Otros

7. ¿Tiene proyectos de desarrollo en curso? ¿Cuántos?

- 6) 1 proyecto en desarrollo
- 7) 2 o más proyectos en desarrollo
- 8) 1 proyecto implementado
- 9) 2 o más proyectos implementados
- 10) Proyectos en desarrollo e implementados

8. Segmento de mercado al que da servicio

- 5) Microempresa
- 6) Pequeña
- 7) Mediana
- 8) Grande

PARTE TÉCNICA

9. Modelos de desarrollo de proyectos utilizados:

- 7) cascada
- 8) espiral
- 9) programación extrema
- 10) incremental
- 11) RUP
- 12) otro

Especifique _____

10. ¿Qué porcentaje del tiempo del proyecto lo dedica a

- 6) análisis
- 7) diseño
- 8) implementación
- 9) pruebas
- 10) puesta en producción

Justifique la respuesta _____

11. ¿Qué porcentaje del costo del proyecto lo destina a

- 5) Gestor de proyecto
- 6) Analistas
- 7) Desarrolladores
- 8) Equipo de pruebas

12. ¿Cómo definiría métrica de software?

13. ¿Qué actividades con control por horas realiza para el desarrollo del proyecto?

- 9) diseñar
- 10) codificar
- 11) probar el sistema
- 12) planear
- 13) corregir errores
- 14) requerimientos
- 15) instalar el sistema
- 16) reuniones de trabajo

14. De los siguientes problemas que se presentan al recibir los requerimientos para el sistema, señale los 3 más comunes

- 7) cliente desconoce el alcance del proyecto
- 8) se reciben requerimientos incompletos
- 9) se reciben requerimientos confusos
- 10) no se mencionan restricciones
- 11) se reciben requerimientos agrupados
- 12) otros

¿Cuáles? _____

15. Tipos de diseño del software

- 5) orientado a objetos
- 6) estructurado
- 7) patrones de diseño
- 8) otros

16. ¿Hace inspección en la fase de diseño?

- 3) si
- 4) no

17. ¿Hace reutilización de software?

- 4) Si
- 5) no
- 6) desconoce

¿Por qué? _____

18. ¿En qué porcentaje estima la reutilización?

- 0-25%
- 25%-50%
- 50%-75%

75%-100%

19. ¿Le sirvió como estrategia? Justifique la respuesta

- 5) 0-25%
 - 6) 25%-50%
 - 7) 50%-75%
 - 8) 75%¿100%
-

20. Personas que prueban el software

- 6) el equipo de desarrollo
- 7) un equipo de desarrollo diferente
- 8) un equipo de pruebas
- 9) el cliente
- 10) otros

Especifique _____

21. Tipos de pruebas

- 8) defectos
- 9) integración
- 10) orientado a objetos
- 11) unitarias
- 12) caja negra
- 13) caja blanca
- 14) stress

22. Identifique 3 factores que afectan el costo del mantenimiento del software.

- 8) rotación de personal
- 9) pruebas insuficientes
- 10) responsabilidad contractual
- 11) falta de documentación
- 12) edad y estructura del programa
- 13) habilidades del personal
- 14) otros

¿Cuáles? _____

23. ¿Corrige los defectos del software al ser reportados?

- 3) Si Total Parcial
- 4) no

24. ¿Registra el tiempo que utiliza en corregir los defectos?

- 3) si
- 4) no

25. ¿Para corregir el mal funcionamiento del software realiza

- 3) reinstalaciones parciales
- 4) reinstalaciones completas

26. Ordene por prioridad: El tipo de mantenimiento es

- 4) perfectivo (mejora la funcionalidad del software)
- 5) correctivo (repara las fallas en el software)
- 1) preventivo

Elaborado por: la autora

Anexo 5: Entrevista para medir la satisfacción del usuario

ENTREVISTA 2 A EMPRESAS DE DESARROLLO DE SOFTWARE

FECHA: _____

OBJETIVO: Conocer el nivel de aceptación de los usuarios con relación al servicio prestado.

DATOS BÁSICOS DE LA EMPRESA:

13. Nombre de la empresa.

14. Cargo que desempeña en la empresa

DATOS DE PRESTACIÓN DE SERVICIO DE SOPORTE

15. ¿Tiene un departamento o área para soporte a usuario?

Si

No

16. ¿Cómo brinda el soporte a usuario? Señale al menos 3 opciones

- Vía telefónica
- Vía Skype
- Vía correo electrónico
- Vía celular
- Vía escritorio remoto
- Visita personal del técnico-analista

17. ¿Registra los requerimientos de los usuarios?

Si

No

18. ¿Con cuántos analistas cuenta para la solución de los requerimientos de los usuarios?

- 1 analista
- 2 analistas
- 3 analistas

- Pésimo (1)

24. ¿Cuáles cree usted que fueron los motivos, en general, que ha hecho que sus clientes prefieran su empresa antes que a otras que ofrecen el mismo servicio? Señale por lo menos 3 motivos

- Por recomendación de otras personas
- Por la cartera de clientes atendidos
- Por la eficiencia de la plataforma adquirida
- Por la eficiencia de la atención al cliente
- Por la eficiencia en el soporte a usuario
- Por los bajos costos en los productos ofertados
- Otros_____

Elaborado por: la autora