



**UNIVERSIDAD DON BOSCO
VICERRECTORÍA DE ESTUDIOS DE POSTGRADO**

**TRABAJO DE GRADUACION
FACTORES HUMANOS EN LA CALIDAD DEL SOFTWARE**

**PARA OPTAR AL GRADO DE
MAESTRO EN ARQUITECTURA DE SOFTWARE**

**ASESOR:
MAESTRO OSCAR QUINTANILLA MENA**

**PRESENTADO POR:
EVA CAMPOS DE BERGANZA**

Antiguo Cuscatlán, La Libertad, El Salvador, Centroamérica.

Septiembre de 2016

Índice

Índice	2
Índice de tablas	7
Índice de figuras	8
Introducción	10
1 Marco conceptual	11
1.1 Calidad	11
1.1.1 Evolución histórica del concepto de calidad	11
1.1.2 Definición de calidad	12
1.1.3 Dimensiones de la calidad	13
1.1.4 Perspectivas de la calidad	14
1.1.5 Importancia de la calidad	14
1.2 Software	15
1.2.1 Definición de software	15
1.2.2 Características del software	15
1.2.3 Tipos de software	15
1.2.4 El proceso de desarrollo de software	15
1.2.5 Elementos que participan en el desarrollo de software	16
1.2.6 Roles en el desarrollo de software	16
1.2.7 Importancia del software	16
1.3 Calidad del software	17
1.3.1 Definición de calidad del software	17
1.3.2 Calidad del producto de software	17
1.3.2.1 Atributos internos	18
1.3.2.2 Atributos Externos	18
1.3.3 Calidad del proceso de software	19
1.3.4 Control de la calidad del software	19
1.3.5 Aseguramiento de la calidad del software	19
1.3.6 Importancia del software de calidad	20
1.4 Factores humanos	20
1.4.1 Definición de factores humanos	20
1.4.2 Factores humanos en el desarrollo de software	21
1.4.3 Factores humanos en otras disciplinas	21
1.4.3.1 Factores humanos en la aviación	21
1.4.3.2 Factores humanos en la salud y seguridad ocupacional	22
1.4.3.3 Factores humanos en los sistemas de calidad	22

1.4.4	Factores humanos en la calidad del software	23
2	Marco metodológico	24
2.1	Tema de investigación	25
2.2	Objetivos	26
2.2.1	Objetivo general	26
2.2.2	Objetivos específicos.....	26
2.3	Preguntas de investigación	26
2.4	Justificación.....	28
2.5	Alcances.....	28
2.6	Limitaciones	29
2.7	Enfoque de la investigación	29
2.8	Estrategias de investigación	30
2.9	Tipo de investigación	30
2.9.1	Fase 1: Investigación cualitativa	31
2.9.2	Fase 2: Investigación cuantitativa	31
2.10	Participantes de la investigación	31
2.10.1	Tipo de muestreo.....	31
2.10.2	Caracterización de la muestra.....	31
2.10.3	Criterios de selección de selección de la muestra	32
2.10.4	Tamaño de la muestra.....	32
2.10.5	Selección de la muestra	33
2.11	Recolección de datos	33
2.11.1	Fuentes de datos.....	33
2.11.2	Técnicas de recolección de datos	33
2.11.2.1	Documentación.....	34
2.11.2.2	Entrevista.....	35
2.11.2.3	Cuestionario	35
2.12	Análisis y organización de datos	36
2.13	Instrumentos de investigación.....	37
2.13.1	Instrumentos para la recolección de datos.....	37
2.13.1.1	Guía de entrevista	37
2.13.1.1.1	Preguntas de la entrevista.....	37
2.13.1.1.2	Protocolo de entrevista.....	38
2.13.1.2	Registro o bitácora de entrevista.....	38
2.13.1.3	Cuestionario	39
2.13.1.3.1	Preguntas del cuestionario.....	39
2.13.1.3.2	Protocolo del cuestionario	41

2.13.2	Instrumentos para la organización de datos	41
2.13.2.1	Sistema de clasificación	41
2.13.2.2	Sistema de valoración	41
2.14	Proceso de investigación	42
2.14.1	Inicio	42
2.14.2	Ejecución	42
2.14.2.1	Revisión de la literatura	42
2.14.2.2	Investigación de campo.....	42
2.14.3	Seguimiento y control	43
2.14.4	Cierre	43
2.15	Entregables	43
2.15.1	Perfil del trabajo de graduación.....	43
2.15.2	Revisión de la literatura	43
2.15.3	Resultados de la investigación de campo.....	44
2.15.4	Tesina	44
2.15.5	Artículo	44
3	Revisión de la literatura.....	45
3.1	Factores humanos en la calidad del software.....	45
3.1.1	Competencias.....	45
3.1.2	Experiencia	49
3.1.3	Profesionalismo	49
3.1.4	Empoderamiento	51
3.1.5	Conciencia de calidad	51
3.1.6	Creatividad	52
3.1.7	Personalidad.....	52
3.1.8	Carácter	54
3.1.9	Estado de ánimo.....	55
3.1.10	Estructura organizativa.....	55
3.1.11	Cultura organizacional.....	56
3.1.12	Estilo de liderazgo	58
3.1.13	Gestión del cambio.....	60
3.1.14	Gestión del conocimiento	60
3.1.15	Trabajo en equipo.....	62
3.1.16	Motivación	67
3.1.17	Satisfacción	67
3.1.18	Sentido de pertenencia.....	68
3.1.19	Propiedad del código.....	69

3.1.20	Entorno ambiental	69
3.2	Clasificación de los factores humanos.....	70
3.3	Buenas prácticas de los factores humanos	71
3.3.1	Del individuo	71
3.3.2	De la organización.....	72
3.3.3	Del involucramiento en el trabajo	74
3.4	Indicadores para la evaluación de los factores humanos.....	75
3.4.1	Del individuo	75
3.4.2	De la organización.....	76
3.4.3	Del involucramiento en el trabajo	78
4	Investigación de campo	79
4.1	Fase 1: Investigación cualitativa	79
4.1.1	Resultados.....	79
4.1.1.1	Participantes.....	79
4.1.1.2	Antecedentes.....	79
4.1.1.3	Contexto del participante.....	80
4.1.1.4	Reconocimiento de los factores humanos que inciden en la calidad del software	81
4.1.1.5	Valoración de los factores humanos que inciden en la calidad del software	81
4.1.2	Análisis y discusión	82
4.2	Fase 2: Investigación cuantitativa	83
4.2.1	Resultados.....	83
4.2.1.1	Participantes.....	83
4.2.1.2	Antecedentes.....	83
4.2.1.3	Contexto del participante.....	86
4.2.1.4	Reconocimiento de los factores humanos que inciden en la calidad del software	89
4.2.1.4.1	Competencias.....	89
4.2.1.4.2	Experiencia.....	90
4.2.1.4.3	Profesionalismo	90
4.2.1.4.4	Empoderamiento	90
4.2.1.4.5	Conciencia de calidad	91
4.2.1.4.6	Creatividad	91
4.2.1.4.7	Personalidad.....	92
4.2.1.4.8	Carácter	93
4.2.1.4.9	Estado de ánimo.....	93
4.2.1.4.10	Estructura organizativa.....	94
4.2.1.4.11	Cultura organizacional.....	94

4.2.1.4.12	estilo de liderazgo.....	95
4.2.1.4.13	Gestión del conocimiento	96
4.2.1.4.14	Tamaño del equipo.....	96
4.2.1.4.15	Distribución del equipo	97
4.2.1.4.16	Cohesión del equipo.....	98
4.2.1.4.17	Motivación	98
4.2.1.4.18	Satisfacción	99
4.2.1.4.19	Sentido de pertenencia	99
4.2.1.4.20	Propiedad del código.....	100
4.2.1.4.21	Entorno ambiental	101
4.2.1.5	Otros factores humanos que inciden en la calidad del software.....	101
4.2.1.6	Valoración de los factores humanos que inciden en la calidad del software	102
4.1	Análisis y discusión	103
5	Conclusiones.....	107
6	Bibliografía	108
7	Glosario	116
8	Siglas.....	120
9	Anexos	121
9.1	Anexo No. 1 – Mapa conceptual.....	121
9.2	Anexo No. 2 – Matriz de congruencia metodológica	122
9.3	Anexo No. 3 – Guía de entrevista.....	123
9.4	Anexo No. 4 – Registro o bitácora de entrevista	127
9.5	Anexo No. 5 – Cuestionario	128
9.6	Anexo No. 6 – Respuestas a la entrevistas.....	128
9.7	Anexo No. 7 – Respuestas al cuestionario	128
9.8	Anexo No. 8 – Resultados del cuestionario	128

Índice de tablas

Tabla No. 1 Evolución histórica del concepto de calidad.....	12
Tabla No. 2 Dimensiones de la calidad.....	13
Tabla No. 3 Perspectivas de la calidad.....	14
Tabla No. 4 Atributos internos de calidad del software.....	18
Tabla No. 5 Atributos externos de calidad del software.....	19
Tabla No. 6 Factores humanos en la aviación.....	22
Tabla No. 7 Criterios de selección de la muestra de expertos.....	32
Tabla No. 8 Uso de la literatura en investigaciones.....	34
Tabla No. 9 Ventajas y desventajas de la técnica de la entrevista.....	35
Tabla No. 10 Preguntas de la entrevista.....	38
Tabla No. 11 Preguntas del cuestionario.....	40
Tabla No. 12 Áreas de conocimiento de la ingeniería de software.....	47
Tabla No. 13 Habilidades en el área de la computación.....	48
Tabla No. 14 Tipos de personalidad.....	53
Tabla No. 15 Visiones del desarrollo de software.....	61
Tabla No. 16 Distribución del tiempo de los desarrolladores.....	63
Tabla No. 17 Características del entorno ambiental de los mejores y peores programadores.....	70
Tabla No. 18 Clasificación de los factores humanos.....	71
Tabla No. 19 Participantes de la investigación cuantitativa.....	79
Tabla No. 20 Participantes de la investigación cuantitativa.....	83
Tabla No. 21 Distribución por tamaño de empresa.....	85
Tabla No. 22 Estilos de personalidad.....	92
Tabla No. 23 Reconocimiento de los factores humanos.....	103
Tabla No. 24 Discusión de los factores humanos.....	105

Índice de figuras

Figura No. 1 Planteamiento del problema.....	24
Figura No. 2 Metodología de investigación.....	25
Figura No. 3 Objetivos específicos y preguntas de investigación.....	27
Figura No. 4 Técnicas de recolección de datos.....	34
Figura No. 5 Instrumentos de recolección de datos.....	37
Figura No. 6 Sistema de clasificación.....	41
Figura No. 7 Proceso de investigación.....	42
Figura No. 8 Problemas causados por no gestionar el cambio.....	60
Figura No. 9 Líneas de comunicación por tamaño de equipo.....	63
Figura No. 10 Tamaño promedio de los equipos de desarrollo.....	65
Figura No. 11 Jerarquía de necesidades humanas.....	68
Figura No. 12 Distribución por sexo.....	84
Figura No. 13 Distribución por edad.....	84
Figura No. 14 Distribución por nivel educativo.....	84
Figura No. 15 Distribución por tipo de empresa.....	85
Figura No. 16 Distribución por sector económico.....	85
Figura No. 17 Distribución por puesto de trabajo.....	86
Figura No. 18 Existencia de unidad o responsable de control o aseguramiento de la calidad.....	86
Figura No. 19 Roles desempeñados en el desarrollo de software.....	86
Figura No. 20 Distribución por años de experiencia.....	87
Figura No. 21 Tamaños de los equipos de desarrollo.....	87
Figura No. 22 Duración de los proyectos.....	87
Figura No. 23 Evaluación de la calidad.....	88
Figura No. 24 Aspectos que evalúa.....	88
Figura No. 25 Técnicas de evaluación.....	89
Figura No. 26 Competencias.....	89
Figura No. 27 Experiencia.....	90
Figura No. 28 Profesionalismo.....	90
Figura No. 29 Empoderamiento.....	91
Figura No. 30 Conciencia de calidad.....	91
Figura No. 31 Creatividad.....	91
Figura No. 32 Personalidad.....	92
Figura No. 33 Carácter.....	93

Figura No. 34 Estado de ánimo.....	93
Figura No. 35 Estructura organizativa.....	94
Figura No. 36 Cultura organizacional.....	94
Figura No. 37 Estilo de liderazgo	95
Figura No. 38 Estilos de liderazgo que inciden	96
Figura No. 39 Gestión del conocimiento	96
Figura No. 40 Tamaño del equipo.....	97
Figura No. 41 Tamaño de equipo óptimo.....	97
Figura No. 42 Distribución del equipo	97
Figura No. 43 Modelos de distribución del equipo	98
Figura No. 44 Cohesión del equipo.....	98
Figura No. 45 Motivación	99
Figura No. 46 Satisfacción	99
Figura No. 47 Sentido de pertenencia	100
Figura No. 48 Propiedad del código.....	100
Figura No. 49 Modelos de propiedad del código	101
Figura No. 50 Entorno ambiental	101
Figura No. 51 Valoración de los factores humanos	102

Introducción

Personas, herramientas y procesos se consideran elementos claves en el desarrollo de software. Con demasiada frecuencia, los aspectos técnicos (herramientas y procesos) se enfatizan más los aspectos humanos (personas). Sin embargo, el desarrollo de software es principalmente un esfuerzo humano. Personas en el papel de usuarios se enfrentan a los problemas que el software tiene como objetivo resolver, definen el problema y los requisitos. Personas en el papel de analistas, diseñadores, arquitectos, programadores y probadores de software analizan los requisitos, especifican y diseñan la solución, producen el código y prueban el software. Personas en el papel de administradores de proyectos gestionan todo el proceso. Y finalmente personas, nuevamente en el papel de usuarios utilizan el software. Son personas las que determinan lo que el software proveerá y la calidad del producto final.

Lo que plantea la pregunta de cuáles son los factores humanos que inciden en la calidad del software. Existen investigaciones que abordan el tema de los factores humanos que inciden en la calidad del software, pero no todos los factores humanos han sido identificados y organizados; y dado el impacto que tienen en el proceso de desarrollo y en el éxito de los proyectos, consideramos que el tema amerita más investigación.

En la investigación que se detalla en este documento, se revisa la literatura para identificar, caracterizar y desarrollar una clasificación de los factores humanos en la calidad del software. Se proporciona un inventario de los factores humanos que han sido estudiados y algunos que no han sido estudiados aún. Además, identifica algunas de las buenas prácticas e indicadores, relacionados con los factores humanos, para apoyar la evaluación de los equipos de desarrollo como parte de los procesos de control y aseguramiento de la calidad del software. Comprende también una investigación mixta que ha establecido el grado de conocimiento y la valoración que de ellos se tiene en el contexto salvadoreño.

Los resultados pueden utilizarse para guiar investigaciones futuras y como punto de partida en la construcción de modelos de calidad para la evaluación de los factores humanos en el desarrollo de software que determinan su calidad.

El documento está organizado en 4 partes: la primera parte presenta el marco conceptual en relación a los términos calidad, calidad de software, control de la calidad, aseguramiento de la calidad y factores humanos; la segunda parte expone la metodología de investigación; la tercera parte presenta la revisión de la literatura relevante; y la última parte, los resultados de la investigación mixta y nuestra discusión y conclusiones sobre los resultados.

1 Marco conceptual

1.1 Calidad

1.1.1 Evolución histórica del concepto de calidad

El concepto de calidad es tan viejo como la humanidad, y a lo largo de la historia han surgido muchas formas de entenderla. Una de las primeras definiciones de calidad, es la idea de calidad como “*excelencia*”, la cual se remonta hasta los filósofos griegos como Platón quien entendía la calidad como “*lo mejor*”. Este concepto de calidad como algo absoluto fue también compartido por Steenkamp y Garvin para quienes significaba el poseer los estándares más altos sin ningún compromiso con lo secundario y oponiéndolo a lo vulgar [1, p. 151].

El concepto técnico de calidad como “*conformidad con las especificaciones*” surgió con Shewhart, para quien la calidad tenía dos aspectos: la **calidad objetiva** o calidad de una cosa como realidad objetiva independiente del hombre, y la **calidad subjetiva** que tiene que ver con lo que pensamos, sentimos o creemos como resultado de la realidad objetiva [2, p. 4]. También afirmó que la definición de calidad debía servir tanto para medir la calidad del producto como para poder hacer un análisis de su evolución en el tiempo [1, p. 154].

La idea de calidad como conformidad con las especificaciones fue retomada por Crosby quien definió la calidad como “*cumplimiento con los requisitos, no elegancia*” y además, la idea de que la única norma de desempeño es “*cero defectos*” [3, p. 110]. Según Crosby las especificaciones o requisitos debían ser claramente definidos por el propio fabricante de manera que no fueran incomprendidos. Esta definición de calidad presenta el inconveniente que las especificaciones al ser definidas por el fabricante no representan necesariamente las necesidades del cliente [1, p. 154].

El concepto estadístico de calidad como “*uniformidad y fiabilidad a bajo costo y adecuado a las necesidades*” surgió de la mano de Deming (1982). Según Deming un producto tiene calidad cuando sus características están dentro de los límites de tolerancia superior e inferior de la especificación, cuando su variabilidad está entre los límites establecidos [1, p. 156]; y cuando ayuda a alguien y goza de un mercado adecuado y sustentable [3, p. 94]. La variabilidad es entonces la principal culpable de la mala calidad y para reducirla es importante centrarse en la mejora continua de los procesos.

Esta definición de Deming es equivalente a la de Wyckoff, para quien la calidad es el grado de excelencia prometido y el control de la variabilidad en el alcance de esa excelencia [1, p. 156]; y congruente con el concepto de Taguchi sobre la función de la pérdida de calidad [1, p. 159].

El concepto de calidad como “*aptitud para el uso*” surgió con Juran [4, p. 2.2], y es la idea predominante en casi todos los enfoques modernos de calidad desde que éste afirmó que el cliente es la pieza más importante de la línea de producción y que la calidad debe orientarse a satisfacer sus necesidades. Según Juran, la calidad se observa cuando el producto satisface las características que le permiten desempeñar la función para la que ha sido diseñado, implicando que el cliente no compra el producto en sí, si no el servicio que le presta el producto [1, p. 162]. Estas ideas han sido compartidas también por Feigenbaum e Ishikawa.

Posteriormente, el concepto de calidad como “*satisfacción de las expectativas del cliente*” fue desarrollado a partir de las ideas de Juran y Deming, para superar dos problemas con la definición anterior: los clientes generalmente no conocen las especificaciones de calidad del producto, y las especificaciones se establecen en los procesos de la cadena de valor pero su percepción solo se forma al final del proceso de compra. Pero, aunque los clientes no conozcan las especificaciones que permiten juzgar la calidad de un modo objetivo, tienen expectativas, que pueden medirse. Proponentes de esta idea son, entre otros, Parasuraman, Zeithaml y Berry [1, p. 118].

Las diferentes definiciones toman distintas unidades de análisis, ponen énfasis en diferentes aspectos, y recurren a distintos métodos para evaluarla.

TABLA NO. 1 EVOLUCIÓN HISTÓRICA DEL CONCEPTO DE CALIDAD

Autor	Definición	Ideas claves	Unidad de análisis	Énfasis	Medición
Platón, Steenkamp (1984), Garvin (1989)	Excelencia	Superioridad absoluta, "lo mejor"	Producto	Calidad de diseño	Abstracta y subjetiva, por experiencia
Shewhart (1931), Crosby (1970)	Conformidad con las especificaciones	Cero defectos	Producto, proceso	Calidad de conformidad	Por comparación contra un estándar
Deming (1982), Wyckoff (1984), Taguchi (1981)	Grado predecible de uniformidad y fiabilidad en torno a un valor objetivo	Reducir la variabilidad, lograr eficacia económica, mejorar los estándares	Producto , proceso	Calidad de diseño, calidad de conformidad	Estadística
Juran (1982), Feigenbaum (1983), Ishikawa (1985)	Aptitud o adecuación para el uso	Las necesidades de los clientes definen las especificaciones	Sistema	Calidad de diseño, calidad de conformidad	Por comparación contra las especificaciones
Parasuraman (1990), Berry (1990), Zeithaml (1990)	Satisfacción de las expectativas del cliente	Alcanzar las expectativas del cliente	Servicio	Calidad de servicio, calidad percibida	Por comparación contra las expectativas del cliente

Nota. Adaptado de *Gestión de la calidad: conceptos, enfoques, modelos y sistemas* [1, p. 147].

1.1.2 Definición de calidad

Hemos visto que existen muchas definiciones de calidad. El hecho es que, no existe en la literatura una definición universal y comúnmente aceptada.

Algunos autores opinan que la calidad no es definible, sino que es una propiedad totalmente personal que solo puede reconocerse a través de la experiencia; o que no es posible llegar a una definición comúnmente aceptada que sea válida para todas las empresas e industrias [1, p. 146]. Kitchenham afirmaba que la calidad es "*difícil de definir, imposible de medir, fácil de reconocer*" [5, p. 3].

Pese a esto, definir lo que entenderemos por calidad es importante, no solo porque de ello depende su relación con otras variables sino también porque consensuar su significado para la investigación permitirá homogenizar el lenguaje y favorecer el diálogo.

Para la investigación, se tomará la definición de Juran, la cual ha sido incorporada en prácticamente la mayoría de las definiciones de calidad adoptadas por organizaciones como la Organización Internacional de Normalización (ISO, por sus siglas en inglés), Asociación Americana para la Calidad (ASQ, por sus siglas en inglés), Asociación Española de Normalización y Certificación (AENOR), Una Norma Española (UNE), Asociación Española para la Calidad (AEC), y Organización Europea para la Calidad (EOQ, por sus siglas en inglés), entre otras.

Juran propuso una definición sencilla de calidad: "*aptitud para al uso*" o "*adaptación al uso*" [3, p. 108]. A esta definición le estableció dos significados subsidiarios [4, pp. 2.1, 2.2]:

- 1) Calidad significa aquellas **características del producto que cumplen con las necesidades del cliente** y por tanto proveen satisfacción al cliente.
- 2) Calidad significa **ausencia de deficiencias** (ausencia errores que requieren hacer el trabajo de nuevo o que resultan en fallas, insatisfacción del cliente, reclamos, etc.)

Una definición congruente con la de Juran es la de la ASQ quien define la calidad como "*las características de un producto o servicio que inciden en su capacidad para satisfacer las necesidades explícitas o implícitas, o un producto o servicio que está libre de deficiencias*" [6, p. 2].

Y también, congruente con la de la ISO 9000 de los “*Sistemas de Gestión de Calidad*”, donde se define la calidad como “*el grado en el que un conjunto de características inherentes cumple con los requisitos*” [7]. En esta definición, **grado** significa que se puede usar el término calidad con adjetivos como mala, buena y excelente; **inherente** significa que existe en algo, en especial como una característica permanente; y **requisito** se refiere a una necesidad o expectativa que se especifica ya sea explícita o implícitamente [6, p. 2].

De estas definiciones pueden extraerse las siguientes ideas [5, p. 3]:

- 1) **La calidad no es absoluta**, significa diferentes cosas en diferentes situaciones y es percibida de forma diferente por diferentes personas.
- 2) **La calidad es multidimensional**, tiene muchos aspectos que contribuyen a ella algunos de los cuales pueden medirse y otros no.
- 3) **La calidad está sujeta a restricciones**, en muchos casos la calidad no puede separarse de los costos, o de otros recursos como las personas, herramientas y el tiempo, los cuales generalmente son limitados.
- 4) **La calidad es acerca de compromisos aceptables**, cuando hay restricciones algunos criterios de calidad deben ser sacrificados por otros.
- 5) **Los criterios de calidad no son independientes**, sino que se interrelacionan unos con otros causando conflictos.

1.1.3 Dimensiones de la calidad

Las dimensiones de la calidad son los aspectos que se tienen en cuenta cuando se evalúa la calidad de un producto o servicio. Éstas fueron definidas por Garvin y son consideradas como independientes, un producto puede ser excelente en una dimensión pero promedio o malo en otra.

TABLA NO. 2 DIMENSIONES DE LA CALIDAD

Dimensión	Significado
Desempeño o rendimiento (<i>Performance</i>)	Características operativas primarias del producto o servicio. Estas características son generalmente medibles, de modo que el producto o servicio puede ser evaluado objetivamente en aspectos individuales de desempeño.
Propiedades, características o prestaciones (<i>Features</i>)	Características secundarias del producto o servicio para mejorar su atractivo al cliente. Estas características complementan sus funciones básicas.
Confiabilidad o fiabilidad (<i>Reliability</i>)	Consistencia de funcionamiento con el paso del tiempo o probabilidad de que el producto no fallará dentro de un período específico de tiempo. Entre las medidas más comunes de la fiabilidad están el tiempo promedio entre fallas y la tasa de errores por unidad de tiempo.
Conformidad o cumplimiento (<i>Conformance</i>)	Precisión con la que el diseño y características del producto o servicio cumplen con las especificaciones.
Durabilidad (<i>Durability</i>)	Mide la duración de vida o vida útil de un producto, incluyendo reparaciones. Tiene tanto una dimensión técnica como económica. Técnicamente significa la cantidad de uso que se obtiene de un producto antes de que se deteriore; y económicamente significa la cantidad de uso que se obtiene de un producto antes de que falle y su reemplazo sea preferible a repararlo.
Servicio o facilidad de servicio (<i>Serviceability</i>)	Velocidad, cortesía, competencia y facilidad de reparación. Velocidad con que el producto puede ser puesto en operación después de fallos, así como la competencia y comportamiento del personal de servicio para dar solución a los problemas y quejas.
Estética (<i>Aesthetics</i>)	Como el producto luce, se siente, suena, sabe o huele. Indica el tipo de respuesta que el usuario tiene al producto basado en su juicio personal y representa las preferencias personales.
Reputación o calidad percibida (<i>Perceived quality</i>)	Calidad percibida atribuida al desempeño en el pasado y otros intangibles como la imagen, circunstancias, publicidad y nombre de la marca.

Nota. Elaboración propia a partir de Control total de la calidad [6, p. 3], Gestión de la calidad [1, p. 180], Administración y control de la calidad [3, p. 165], Competing on the eight dimensions of quality [8] y What does “product quality” really mean? [9].

1.1.4 Perspectivas de la calidad

Garvin también identificó 5 perspectivas o facetas de la calidad a partir de las cuales ésta puede ser definida y entendida.

TABLA NO. 3 PERSPECTIVAS DE LA CALIDAD

Perspectiva	Significado
Trascendental (<i>Transcendental</i>)	También denominada "calidad relativa" relaciona la calidad con la excelencia innata y hace referencia al hecho de que la calidad es fácil de percibir y reconocer, pero difícil de definir. La calidad es un ideal al que se intenta llegar. Todos tenemos una visión de lo que constituye un producto o servicio de calidad.
Del usuario (<i>User based</i>)	Hace referencia a la calidad como conformidad con aquello que el cliente espera recibir y que fue establecido en las especificaciones. Adecuación o adaptación al propósito.
Del proceso (<i>Manufacturing based</i>)	Hace referencia a la conformidad con los requisitos o especificaciones. Identifica la calidad del producto con la calidad del proceso de producción y la entrega del producto. Todo producto fabricado de acuerdo a estándares regulados de calidad podrá ser considerado un producto de calidad.
Del producto (<i>Product based</i>)	Relaciona la calidad con ciertas características internas del producto. Mientras que las vistas del usuario y del proceso definen la calidad "desde fuera", la del producto la define "desde dentro".
Del valor (<i>Value based</i>)	La calidad depende de lo que el cliente esté dispuesto a pagar. Establece una relación entre la cantidad de dinero que el cliente está dispuesto a pagar y la calidad del producto.

Nota. Elaboración propia a partir de Ingeniería de Software [10, p. 383], Software Quality [5, p. 13], [2, p. 8] y What does "product quality really mean? [9]

1.1.5 Importancia de la calidad

Entendiendo la calidad como cumplir con las necesidades del cliente, puede afirmarse que la calidad es importante porque permite [4, p. 2.2]:

- 1) Incrementar y mantener la satisfacción y lealtad del cliente.
- 2) Hacer productos más vendibles.
- 3) Alcanzar a la competencia.
- 4) Incrementar la cuota de mercado.
- 5) Incrementar las ventas.
- 6) Asegurar el precio de los productos.

El mayor efecto es en las ventas, y bajo esta óptica, en términos de costos, mayor calidad cuesta más.

Ahora bien, entendiendo la calidad como la ausencia de deficiencias, puede afirmarse que la calidad es importante porque permite [4, p. 2.2]:

- 1) Reducir la insatisfacción del cliente.
- 2) Reducir la tasa de errores en la fabricación.
- 3) Reducir la necesidad de rehacer el trabajo.
- 4) Reducir la necesidad de inspecciones y pruebas.
- 5) Reducir las fallas en la operación.
- 6) Reducir el tiempo de entrega del producto.
- 7) Mejorar la eficiencia en la entrega.
- 8) Reducir los costos.

El mayor efecto es en los costos, y bajo esta óptica, en términos de costos, mayor calidad cuesta menos.

1.2 Software

1.2.1 Definición de software

El término software suele atribuirse a Tukey por su artículo publicado en 1957, donde definía software como “*programa de computador*” o conjunto de instrucciones para que un computador lleve a cabo una tarea [10, p. 13].

Esta definición es, en la actualidad, incompleta pues el software no solo comprende los programas sino también los documentos que los describen y cualquier otro artefacto relacionado con el mismo. Una definición más completa es la siguiente:

“El software es: a) instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados; b) estructuras de datos que permiten que los programas manipulen la información, y c) información descriptiva tanto en papel como en formas virtuales que describe la operación y uso de los programas” [11, p. 3].

1.2.2 Características del software

El software tiene características que lo hacen diferente de cualquier otro producto [11, p. 4] [5, p. 7] [10, p. 14]:

- 1) No tiene existencia física.
- 2) No se puede representar completamente en diagramas.
- 3) Se desarrolla o modifica con intelecto, no se manufactura en el sentido clásico.
- 4) La falta de conocimiento de las necesidades del cliente al inicio del proceso de desarrollo.
- 5) Es propenso al cambio, debe adaptarse a nuevos requisitos o necesidades del cliente a lo largo del tiempo.
- 6) Los rápidos cambios tecnológicos en el hardware (donde el software se ejecuta) y en el software mismo (utilizado para desarrollar software).
- 7) No se “desgasta”, la tasa de fallas no aumenta conforme pasa el tiempo debido a su desgaste por el uso.
- 8) La mayor parte se construye para un uso individualizado. Esto es cierto aun cuando en la actualidad se ha iniciado con el desarrollo de software basado en componentes reusables.
- 9) Las altas expectativas de los clientes, especialmente en relación con su adaptabilidad.

1.2.3 Tipos de software

En términos generales existen dos tipos de productos de software [12, p. 6]:

- 1) **Software genérico**, sistemas independientes que son desarrollados por una empresa de software y se venden en el mercado abierto a cualquier cliente que desee comprarlo. Ejemplos de estos productos son: procesadores de texto, hojas electrónicas, sistemas gestores de base de datos, paquetes para diseño gráfico, Sistemas de Planeamiento de los Recursos Empresariales (ERP, por sus siglas en inglés), Sistemas para la Gestión de la Relación con el Cliente (CRM, por sus siglas en inglés), etc.
- 2) **Software personalizado o a la medida**, sistemas que están destinados para un cliente en particular.

1.2.4 El proceso de desarrollo de software

El proceso de desarrollo de software o proceso de software es un conjunto coherente de políticas, estructuras organizativas, tecnologías, procedimientos, actividades y artefactos que se necesitan para concebir, desarrollar, implantar y mantener un producto de software [10, p. 22].

Una definición más sencilla se refiere al proceso de software como la secuencia de actividades que conducen a la elaboración de un producto de software [12, p. 28].

Existen cuatro actividades fundamentales que son comunes a todos los procesos de software [12, p. 28]:

- 1) **Especificación del software**, donde el cliente y los responsables de desarrollarlo definen los requisitos y diseñan el software que se desarrollará y las restricciones en su operación.
- 2) **Desarrollo del software**, donde se diseña y programa el software.
- 3) **Validación o pruebas del software**, donde se verifica el software para asegurar que provea lo que el cliente requiere.
- 4) **Evolución o mantenimiento del software**, donde se modifica el software para reflejar los requisitos cambiantes del negocio que el software soporta.

1.2.5 Elementos que participan en el desarrollo de software

En el desarrollo de software participan tanto elementos tecnológicos (hardware, software de sistemas y de bases de datos, lenguajes de programación, mecanismos de interconexión, enfoques de desarrollo y de gestión de proyectos, etc.) como humanos (clientes o usuarios, directores de proyectos, arquitectos de sistemas, analistas, desarrolladores, etc.) y los factores humanos que se derivan de ellos.

De estos, la importancia del recurso humano, especialmente del equipo de desarrollo, y los factores humanos que se derivan de ellos están siendo reconocidos como los más importantes. Se ha llegado a afirmar que los factores humanos son el origen de la mayor parte de los problemas en el desarrollo de software: dificultades de comunicación, no contar con el recurso humano idóneo, alta rotación de personal, falta de motivación, empleados insatisfechos, equipos no integrados, etc. [13, p. 7] [14, pp. 74-76].

1.2.6 Roles en el desarrollo de software

Los roles y responsabilidades en el desarrollo de software varían de acuerdo al enfoque de proceso de desarrollo de software y al enfoque de gestión elegido, pero pueden en términos generales agruparse en los siguientes [15, pp. 2-3]:

- 1) Los **patrocinadores** del proyecto, son las personas que proveen los fondos para el desarrollo de software.
- 2) Los **usuarios**, son los que enfrentan el problema que el software busca resolver, definen el problema y los requisitos que el software debe satisfacer, y son los que utilizan el software.
- 3) El **equipo de desarrollo**, son los que bajo roles específicos (analistas, diseñadores, arquitectos, administradores de proyecto, programadores, probadores, etc.) analizan los requisitos, especifican y diseñan la solución de software, producen el código y prueban el software.

1.2.7 Importancia del software

La importancia del software puede establecerse de muchas maneras. En primer lugar, el software afecta casi todos los aspectos de nuestras vidas, está presente en casi cualquier actividad cotidiana, en el comercio y la cultura, por ejemplo [11, p. 718]. Y en segundo lugar, el software genera el artículo más importante que cualquier individuo u organización puede adquirir: la **información**.

El software transforma los datos de modo que puedan ser más útiles en un contexto específico, gestiona la información de negocios para mejorar la competitividad, provee una vía para las redes de información y brinda los medios para obtener información en todas sus formas [11, pp. 1, 3].

En la actualidad, la industria del software se ha convertido en un factor dominante [11, p. 3]. Este hecho se evidencia con el crecimiento que ha tenido en los últimos años. En los Estados Unidos, en el 2007, la industria del software tuvo un crecimiento anual del 14%, comparado con un 2% para todas las industrias [16]. También, en el 2012, la industria del software tuvo una contribución de más del 20% al Producto Interno Bruto (PIB) [17].

Aunque las cifras indicadas describen la situación de los Estados Unidos, se puede esperar un comportamiento comparable en el resto de los países del mundo. Para el caso, en América Latina un estudio de la CEPAL, realizado en el 2012, indica que en algunos países latinoamericanos más del 50% de la población tiene acceso a Internet [18]. Esta cifra no se refiere al consumo de software en particular pero muestra la importancia que está adquiriendo el software en las sociedades hoy en día, al proveer la vía sobre la cual la Internet opera. En El Salvador, desde diciembre de 1995, cuando el país inició con acceso a Internet, hasta el 2013, un 20% de los salvadoreños tenían acceso a Internet [19]; y además, en el 2013 El Salvador se posicionó como el tercer país en Centro América con mayor cantidad de usuarios en *Facebook*, con más de 1.5 millones de perfiles [20].

Puede afirmarse entonces que, el mundo depende del software, especialmente el software de calidad.

1.3 Calidad del software

1.3.1 Definición de calidad del software

En cuanto a la definición de calidad de software, se tiene la definición de la IEEE:

- “a) El grado en que un sistema, componente o proceso cumple con los requisitos especificados.*
- b) El grado en que un sistema, componente o proceso cumple con las necesidades o expectativas del cliente o usuario” [21].*

Esta definición es congruente con la definición de quien es considerado el “*Padre de la Calidad del Software*”, Humphrey Watts. Según Humphrey, el software de calidad es aquel que:

- “a) Cumple con las necesidades funcionales del usuario, con la fiabilidad y consistencia de realizar el trabajo del usuario.*
- b) No tiene defectos (algo que está mal en el programa, en el diseño, en sus requisitos, especificaciones, o en la documentación)” [22].*

A su vez, la guía SWEBOK en su versión 3.0 indica que la calidad del software se refiere a:

- “a) Las características deseadas de los productos de software,*
- b) El grado en que un producto de software particular posee dichas características,*
- c) Los procesos, herramientas y técnicas usados para alcanzar esas características” [23, pp. 10-1].*

Esta definición distingue los dos aspectos de la calidad: la **calidad del producto** de software y la **calidad del proceso** de desarrollo de software [10, p. 375].

1.3.2 Calidad del producto de software

“Un producto es de buena calidad si le sirve a quien lo adquiere y si este lo usa para realizar las tareas para lo que fue concebido” [24, p. 35].

Como se mencionó en las perspectivas de la calidad, la calidad del producto apunta a sus características y viene definida como el grado de observancia de un conjunto de criterios o atributos

de calidad preestablecidos y medibles [10, p. 384]. El software tiene tanto atributos de calidad internos como externos [25, p. 463].

1.3.2.1 ATRIBUTOS INTERNOS

Los atributos internos de calidad del software son de importancia para los ingenieros de software y desarrolladores. Conducen a un software de alta calidad desde el punto de vista técnico [11, p. 94]. Son atributos internos de calidad del software: la mantenibilidad, la portabilidad, la flexibilidad, la reusabilidad, la legibilidad, el poder ser probado y el ser comprensible [25, pp. 463-465].

TABLA NO. 4 ATRIBUTOS INTERNOS DE CALIDAD DEL SOFTWARE

Atributo	Significado
Mantenibilidad (<i>Maintainability</i>)	Facilidad con la cual un software puede ser modificado para cambiar o añadir funcionalidades, mejorar el desempeño o corregir defectos. Disposición para ser modificado para ser corregido, adaptado o ampliado.
Flexibilidad (<i>Flexibility</i>)	La medida con la que se puede modificar un sistema para usos diferentes para los cuales fue diseñado. Capacidad para introducir cambios en función de las necesidades del negocio.
Portabilidad (<i>Portability</i>)	Facilidad para usar el software en un nuevo entorno (sistemas operativos, bases de datos, etc.)
Reusabilidad (<i>Reusability</i>)	La medida con la que y la facilidad con que pueden usarse partes de un sistema en otro sistema.
Legibilidad (<i>Readability</i>)	La facilidad con la que puede ser leído y entendido el código fuente de un sistema.
Poder ser probado (<i>Testability</i>)	El grado en que se pueden hacer pruebas unitarias o de sistema. Capacidad de validar los requisitos establecidos para el software.
Comprensible (<i>Understandability</i>)	Facilidad para entender el propósito y estructura del software.

Nota. Elaboración propia a partir de Code Complete [25, pp. 463-465], Ingeniería de Software [10, pp. 384-392] y Software Quality [5, pp. 17-32].

1.3.2.2 ATRIBUTOS EXTERNOS

Los atributos externos de calidad del software son aquellas propiedades del software fácilmente observables por los usuarios [11, p. 94]. Los atributos externos son las únicas características que interesan al usuario. Son atributos externos de calidad del software: la corrección, la usabilidad, la eficiencia, la fiabilidad, la integridad, la adaptabilidad, la exactitud y la robustez [25, pp. 463-465].

TABLA NO. 5 ATRIBUTOS EXTERNOS DE CALIDAD DEL SOFTWARE

Atributo	Significado
Corrección (<i>Correctness</i>)	El grado en que el sistema está libre de errores en su especificación, diseño e implementación. Si fue construido correctamente. Que las funcionalidades solicitadas en su especificación se encuentren disponibles.
Usabilidad (<i>Usability</i>)	La facilidad con la que los usuarios pueden aprender a usar y usar el sistema.
Eficiencia (<i>Efficiency</i>)	Mínimo uso de recursos de sistema durante su funcionamiento, incluyendo memoria y tiempo de ejecución.
Fiabilidad (<i>Reliability</i>)	La habilidad del sistema de ejecutar su funcionalidad bajo condiciones establecidas cuando sea requerido. Tener un amplio tiempo medio entre fallas, ausencia de defectos.
Integridad (<i>Integrity</i>)	El grado en que el sistema previene el acceso no autorizado o inapropiado a sus programas y sus datos. La idea de integridad incluye el restringir el acceso no autorizado así como asegurar que la información sea usada apropiadamente.
Adaptabilidad (<i>Adaptability</i>)	La medida en que el sistema puede ser usado, sin modificación en aplicaciones o ambientes diferentes de aquel para el cual fue diseñado.
Exactitud (<i>Accuracy</i>)	El grado en el cual el sistema, tal como fue construido, está libre de errores, especialmente con respecto a resultados cuantitativos. Que tan bien el sistema hace el trabajo para el cual fue diseñado.
Robustez (<i>Robustness</i>)	El grado en que el sistema continúa funcionando ante la presencia de entradas incorrectas o condiciones adversas.

Nota. Elaboración propia a partir de Code Complete [25, pp. 463-465], Ingeniería de Software [10, pp. 384-392] y Software Quality [5, pp. 17-32].

1.3.3 Calidad del proceso de software

“Un proceso malo, mal concebido e implementado generará productos de mala calidad. Un proceso bueno, bien concebido e implementado generará mayor cantidad de las veces productos de buena calidad” [24, p. 35].

La calidad del proceso identifica la calidad con la calidad del proceso de producción. Todo producto fabricado de acuerdo a estándares regulados de calidad podrá ser considerado un producto de calidad [10, p. 383].

1.3.4 Control de la calidad del software

El control de la calidad es el proceso que tiene por objetivo identificar los defectos de un producto y proveer la confianza de su corrección y exactitud [26, p. 7]. Esto es, garantizar la calidad del producto e identificar y corregir los defectos en los productos actualmente desarrollados. Es un proceso **reactivo**.

El control de la calidad del software involucra inspecciones o revisiones formales que examinan críticamente cada entregable (documentación, código fuente, etc.), y pruebas frecuentes y extensas.

1.3.5 Aseguramiento de la calidad del software

La IEEE define el aseguramiento de la calidad del software como el patrón planeado y sistemático de todas las acciones necesarias para proveer adecuada confianza de que el software se adecúa o cumple con los requisitos especificados [26, p. 30]. El aseguramiento de la calidad tiene por objetivo garantizar la calidad en el proceso mediante el cual el producto es desarrollado y prevenir defectos. Es un proceso **proactivo**.

1.3.6 Importancia del software de calidad

Resulta innegable que el software es importante, y especialmente el software de calidad. La calidad resulta ser una característica crítica pues da al software su valor comercial y su ausencia se traduce, entre otras cosas, en pérdidas financieras para la organización así como usuarios insatisfechos [5, p. 9].

Existen muchas historias de desastres cuya causa reside en la calidad del software, en la forma de errores en el código. Algunos ejemplos son: la pérdida del *Mariner 1* (Estados Unidos) en 1962 por el uso de un punto en lugar de una coma; la aplicación de dosis letales de terapia radiactiva con la máquina *Therac 25* (Canadá) en 1985 por error en el código; la caída de la red de larga distancia de AT&T (Estados Unidos) en el año de 1990 por una línea de código con error; la falla del Misil *NIM-104 Patriot* al interceptar un misil iraquí (Estados Unidos) en el año de 1991 por un error de aproximación; y la explosión del satélite *Ariane 5* (Europa) en 1996 por mala elección de la precisión de una variable [27] [28].

En los ejemplos citados, los productos de software no lograron alcanzar la calidad esperada. Cabe preguntarse entonces, ¿cómo se logra un software de calidad? La respuesta inmediata es: asegurando que cada uno de los elementos que intervienen en su desarrollo, el producto de software y el proceso de desarrollo sean de calidad. Cada etapa del desarrollo de software debe centrarse en la calidad, la condición de salida para toda actividad, acción y tarea del proceso debe centrarse en la calidad del producto [11, p. 84].

1.4 Factores humanos

1.4.1 Definición de factores humanos

El término factores humanos ha sido identificado en muchas disciplinas, a menudo con acepciones diferentes. Sin embargo, en términos generales puede decirse que los factores humanos son todos aquellos aspectos que pueden influir en las personas y su comportamiento [29].

En un contexto de trabajo, factores humanos son los factores del medio ambiente, de la organización y de las interacciones con otros, y las características individuales que influyen en el comportamiento en el trabajo [29].

Entre estos factores relacionados con el medio ambiente, la organización y las interacciones con otros se mencionan las actitudes hacia el trabajo o enunciados (favorables o desfavorables) acerca de objetos, individuos o eventos, y se refieren a la satisfacción laboral, el involucramiento en el trabajo y el compromiso organizacional [30, p. 74].

Entre las características individuales se tienen, en primer lugar, los factores relacionados con las características físicas o biográficas como la edad, género, raza, discapacidades y antigüedad en el empleo [30, p. 44], las características psicológicas como la personalidad y el carácter [30, p. 133], y las relacionadas con las aptitudes o capacidad que tiene el individuo para llevar a cabo las diferentes tareas de un puesto de trabajo.

Las aptitudes generales de un individuo están constituidas por dos conjuntos de factores: intelectuales y físicos [30, p. 52]. Las aptitudes o capacidades intelectuales son las habilidades que se necesitan para desarrollar actividades mentales como pensar, razonar y resolver problemas [30, p. 52]; y las aptitudes o capacidades físicas se refieren a la capacidad para realizar tareas que demandan fuerza en la forma de resistencia dinámica, corporal, estática o explosiva; flexibilidad en la forma de estiramiento o movimiento; y otros aspectos como coordinación, equilibrio y vigor [30, p. 55].

1.4.2 Factores humanos en el desarrollo de software

El desarrollo de software es una actividad principalmente humana. El software es especificado y construido por personas o grupos de personas en una organización, y como tal, se ve influenciado principalmente por personas. La inversión en la mejora de procesos, metodologías y técnicas es de poca importancia cuando el empleado aplicándolas no tiene los conocimientos y habilidades apropiadas, o no tiene interés en utilizarlas [31, p. 47].

Este hecho se confirma considerando que los principales aspectos que se han identificado que garantizan el éxito de los proyectos de desarrollo de software son de naturaleza humana [24, p. 37]:

- 1) Involucramiento del usuario.
- 2) Soporte e involucramiento de los puestos directivos.
- 3) Establecimiento de requisitos claros.
- 4) Planeamiento apropiado.
- 5) Expectativas realistas.
- 6) Hitos más cortos o pequeños.
- 7) Recurso humano competente.
- 8) Apropiación.
- 9) Visión y objetivos claros.
- 10) Recurso humano trabajador y enfocado.

Y sin embargo, aún con la importancia del recurso humano en el desarrollo de software no se ha llegado a establecer una caracterización completa de los mismos como se ha hecho en otras disciplinas.

1.4.3 Factores humanos en otras disciplinas

1.4.3.1 FACTORES HUMANOS EN LA AVIACIÓN

Los factores humanos han sido bastante estudiados y caracterizados en la aviación, pues se reconoce que el error humano más que las fallas mecánicas es la causa de la mayoría de los accidentes e incidentes.

En esta disciplina los factores humanos se refieren a los "*aspectos que influyen a las personas de forma diferente por no tener todos las mismas capacidades, fortalezas, debilidades y limitaciones*" [32, pp. 14-3].

El mismo término también se aplica a la ciencia que los estudia, y se define como "*la ciencia del conocimiento de las propiedades de las capacidades humanas, la aplicación de ese conocimiento al diseño, desarrollo e implementación de sistemas y servicios, y el arte de asegurar la aplicación exitosa de los principios de factores humanos en el ambiente de trabajo del mantenimiento*" [32, pp. 14-3]. Como ciencia, es multidisciplinaria y comprende la psicología clínica, psicología experimental, antropometría, ciencias de la computación, ciencia cognitiva, ingeniería de la seguridad, ciencia médica, psicología organizacional, psicología educacional, e ingeniería industrial [32, pp. 14-3 a 14-6].

Para caracterizarlos se utiliza el modelo PEAR que clasifica los factores humanos en 4 categorías [32, pp. 14-8 a 14-10]:

- 1) Personas (**P**eople): las personas que hacen el trabajo.
- 2) Ambiente (**E**nviroment): el ambiente en el cual trabajan.
- 3) Acciones (**A**ctions): las acciones que realizan.
- 4) Recursos (**R**esources): los recursos necesarios para realizar el trabajo.

En la siguiente tabla se muestran los factores humanos considerados en cada una de estas categorías:

TABLA NO. 6 FACTORES HUMANOS EN LA AVIACIÓN

Categoría	Factor humano
Personas (People)	Físicos: tamaño físico, sexo, edad, fuerza, limitaciones sensoriales.
	Fisiológicos: carga de trabajo, experiencia, conocimiento, entrenamiento, actitud, estado mental y emocional.
	Psicológicos: factores nutricionales, salud, estilo de vida, fatiga, dependencia química.
	Psicosociales: conflictos interpersonales.
Ambiente (Environment)	Físicos: clima, localización dentro y fuera, espacio de trabajo, horario, iluminación, nivel de sonido, seguridad.
	Organizacionales: empleados o compañeros de trabajo, supervisión, relaciones fuerza de trabajo-administración o gerencia, presiones, estructura de la tripulación, tamaño de la compañía, rentabilidad, moral, cultura corporativa.
Acciones (Actions)	Pasos para realizar una tarea, secuencia de actividades, número de personas involucradas en la actividad, requisitos de control de la información, requisitos de conocimiento, requisitos de habilidad, requisitos de actitud, requisitos de certificación, requisitos de inspección.
Recursos (Resources)	Tarjetas de procedimientos o trabajo, manuales técnicos, papeleo y autorizaciones, otras personas, equipos de prueba, herramientas, accesorios, materiales, computadores, equipo para iluminación de la tarea, software, técnicas de operación en tierra, puestos de trabajo y ascensores, formación, sistemas de calidad.

Nota. Elaboración propia a partir de *Aviation Maintenance Technician Handbook* [32, pp. 14-8 a 14-10].

1.4.3.2 FACTORES HUMANOS EN LA SALUD Y SEGURIDAD OCUPACIONAL

El enfoque de la salud y seguridad ocupacional de los factores humanos tiene por objetivo reducir el error e influenciar el comportamiento, y los define como “*los aspectos ambientales, organizacionales y del trabajo, y las características humanas e individuales, que influyen el comportamiento en el trabajo en una forma que puede afectar la salud y la seguridad*” [33, p. 2].

La definición comprende 3 categorías de factores humanos [33, p. 2]:

- 1) **Del trabajo:** el trabajo debe ser diseñado de acuerdo a principios ergonómicos que tomen en cuenta las limitaciones y fortalezas humanas e incluye factores como la naturaleza de la tarea, la asignación al individuo apropiado, la carga de trabajo, el ambiente de trabajo, el diseño de indicadores y controles, los procedimientos, y aspectos mentales como los requisitos perceptivos, de atención y de toma de decisiones.
- 2) **Del individuo:** características individuales que influyen el comportamiento en formas complejas como las competencias, habilidades, personalidad, actitud y percepción del riesgo.
- 3) **De la organización:** aspectos de la organización que pueden influenciar el comportamiento individual y grupal como los patrones de trabajo, cultura organizacional, recursos, comunicaciones, liderazgo, etc.

1.4.3.3 FACTORES HUMANOS EN LOS SISTEMAS DE CALIDAD

En la norma ISO 9001 de los “*Sistemas de Gestión de Calidad*” se identifican los factores humanos que se originan en las competencias, como: la educación, la formación, las habilidades y la experiencia [7, p. 7].

A su vez, la norma ISO 10018 en sus “*Lineamientos sobre el involucramiento y competencias de las personas en la Gestión de la Calidad*” identifica otros factores humanos: los asociados a las competencias, como: el reclutamiento, la educación, la formación, el empoderamiento, la responsabilidad, la conciencia de calidad, la creatividad e innovación; los derivados de las interacciones de las personas con la organización: las líneas de responsabilidad, las líneas de autoridad y el liderazgo; y finalmente los que se originan en las relaciones con otros individuos al

realizar el trabajo: el trabajo en equipo, la colaboración, la actitud, la motivación, la comunicación, el compromiso, las redes de relaciones y los reconocimientos [34].

1.4.4 Factores humanos en la calidad del software

Aunque existe cierta tendencia a enfatizar los procedimientos, herramientas y sistemas como determinantes de la calidad del software, la calidad es acerca de las personas:

“Son personas y las organizaciones conformadas por personas los que tienen los problemas a ser abordados por el software, son personas las que definen el problema y especifican la solución, son personas las que implementan el diseño y producen el código, son personas las que prueban el código, son personas las que usan el sistema final y las que harán juicios acerca de la calidad de toda la solución” [5, p. 15].

De hecho, las principales causas de la mala calidad del software tienen su origen en el factor humano: desconocimiento del negocio, desconocimiento de la tecnología, plazos irreales, mala ingeniería de software y malas decisiones de adquisición [35].

Existen algunos estudios que identifican individualmente algunos factores humanos y que confirman o no su incidencia en la calidad del software en contextos particulares.

Algunos ejemplos de estos estudios son: los citados por McConnell [25, pp. 682, 820] que describen la influencia de las competencias individuales en la calidad del software; Sackman, Erickson y Grant [25, p. 682] quienes describen la relación de la experiencia y la calidad del software y la productividad; Mathew [36] que describe el impacto de los factores organizacionales en la calidad del software; Nagappan, Murphy y Basili [37] en un estudio patrocinado por Microsoft establecen la relación entre la estructura organizativa, la cohesión del equipo y el número de personas modificando el código con la calidad del software; Lavallée y Robillard [36] que identifican aspectos culturales que inciden negativamente en la calidad del software; Jaktman [36] que describe la relación entre los valores organizacionales y la calidad del software; Prifling [38] que explora los estilos de liderazgo que promueven la calidad en proyectos de desarrollo; Basili y Caldiera [39] quienes proponen las “fábricas de experiencia” para mejorar la calidad del software; Meneely y Williams [40] y Mahajan [41] que establecen una posible relación entre el tamaño del equipo y la calidad; y un estudio de Bird, Nagappan, Devanbu, Gall y Murphy [42] realizado conjuntamente por la Universidad de California, la Universidad de Zurich y Microsoft que evalúa la tasa de errores en el desarrollo de Windows Vista en equipos distribuidos y equipos locales.

2 Marco metodológico

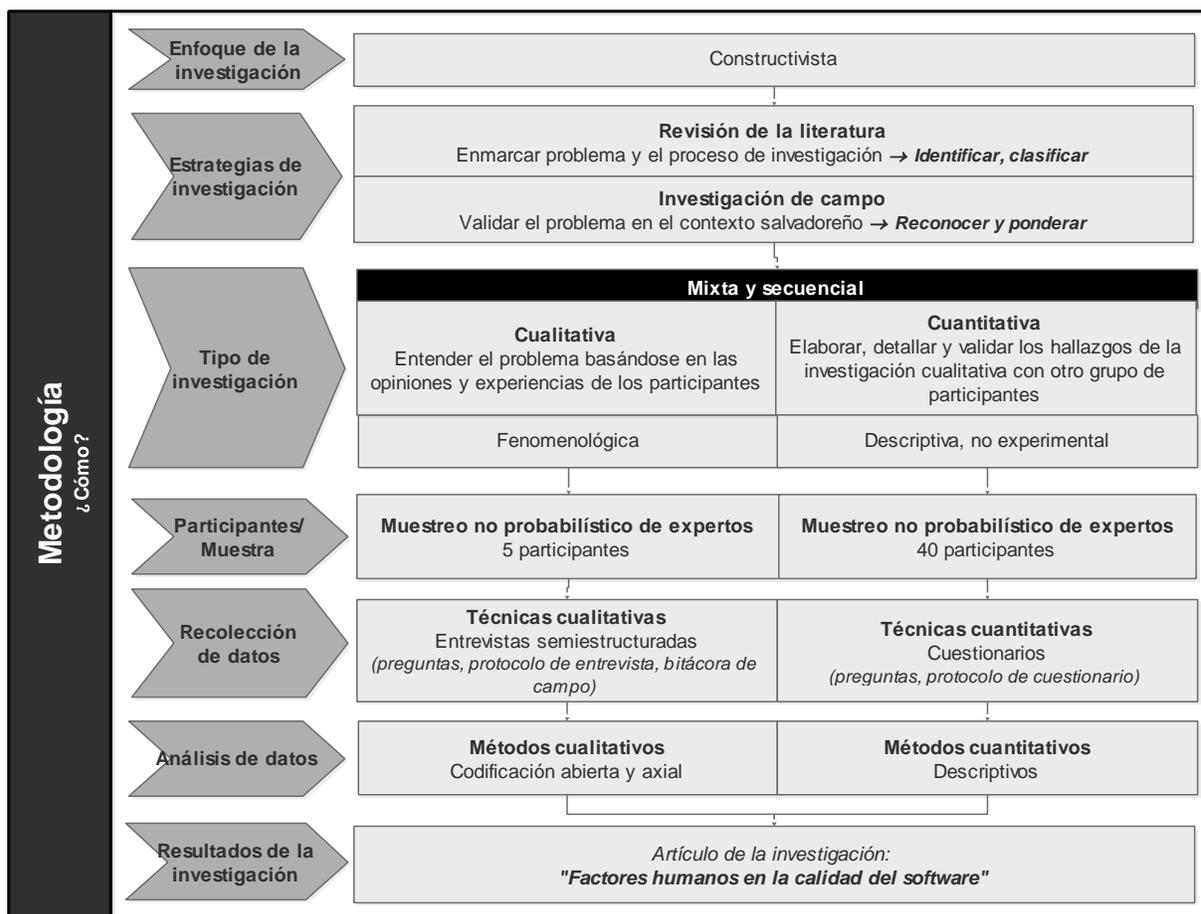
A continuación se presenta un esquema del planteamiento del problema y de la metodología de investigación que se describe en este documento.

FIGURA NO. 1 PLANTEAMIENTO DEL PROBLEMA



Nota. Elaboración propia.

FIGURA NO. 2 METODOLOGÍA DE INVESTIGACIÓN



Nota. Elaboración propia.

Véase en el anexo No. 2 la matriz de congruencia metodológica.

2.1 Tema de investigación

La investigación tiene por propósito estudiar los factores humanos que inciden en la calidad del software. El **enunciado del tema de investigación** es:

"Factores humanos en la calidad del software".

La investigación se desarrolla en dos etapas. La primera, una **revisión de la literatura** que a partir del **estudio de investigaciones primarias, secundarias y terciarias** permite **identificar y caracterizar los factores humanos que inciden en la calidad del software**; y la segunda, una **investigación de campo** que a partir de **entrevistas y cuestionarios** con profesionales que trabajan o han trabajado en el desarrollo de software en El Salvador permite **contextualizar o reconocer los factores humanos que inciden en la calidad del software en el contexto salvadoreño**.

Los aportes que la investigación provee son:

- 1) Inventario comentado de los factores humanos que inciden en la calidad del software.
- 2) Bibliografía de investigaciones primarias sobre los factores humanos y la calidad del software.
- 3) Inventario de buenas prácticas, relacionadas con los factores humanos identificados, que posibilitan el desarrollo de software de calidad.

- 4) Lista de comprobación de parámetros o indicadores, para los factores humanos identificados, que puede ser utilizada para el control y aseguramiento de la calidad del software.
- 5) Clasificación de los factores humanos identificados.
- 6) Análisis de los factores humanos que se reconocen en el contexto salvadoreño y el grado de importancia que se les asigna.

2.2 Objetivos

2.2.1 Objetivo general

La investigación tiene por **objetivo general**:

“Identificar y caracterizar los factores humanos que inciden en la calidad del software desde la perspectiva del control y aseguramiento de la calidad”.

La investigación permitirá comprender los factores humanos que inciden en la calidad del software, reconocer las buenas prácticas que posibilitan el desarrollo de software de calidad, y algunos parámetros o indicadores para evaluar los factores humanos como parte de los procesos de control y aseguramiento de la calidad del software.

2.2.2 Objetivos específicos

Para lograr el propósito del objetivo general se establecen los siguientes **objetivos específicos**:

- 1) Identificar los factores humanos que, según la revisión de la literatura, inciden en la calidad del software.
- 2) Identificar las buenas prácticas, relacionadas con los factores humanos identificados, que posibilitan el desarrollo de software de calidad.
- 3) Identificar algunos parámetros o indicadores, en los factores humanos identificados, que podrían utilizarse en el control y aseguramiento de la calidad del software.
- 4) Desarrollar una clasificación de los factores humanos identificados que inciden en la calidad del software.
- 5) Reconocer los factores humanos que según la perspectiva y experiencia de los profesionales en el desarrollo de software en El Salvador inciden en la calidad del software.
- 6) Determinar el grado de importancia que los profesionales en el desarrollo de software en El Salvador dan a los factores humanos que inciden en la calidad del software.

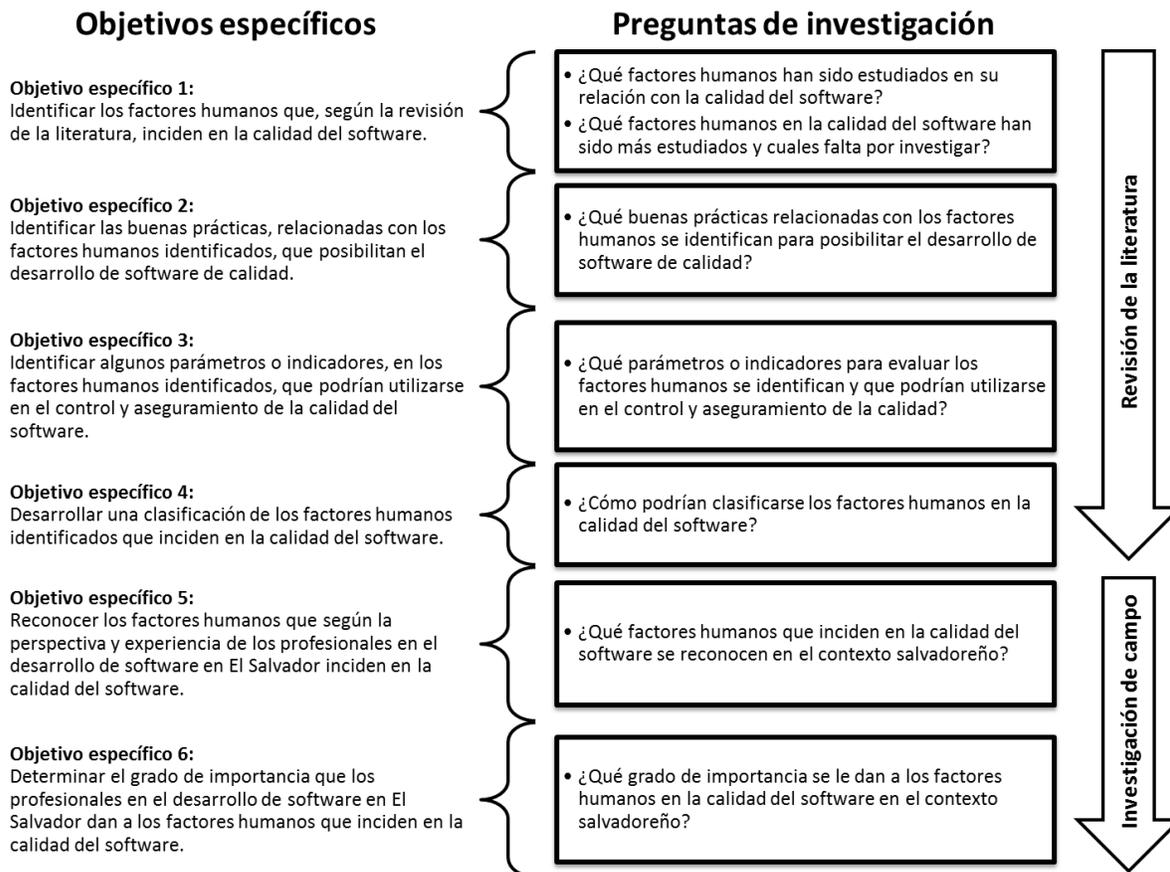
2.3 Preguntas de investigación

Las preguntas a las que la investigación intentará dar respuesta son:

- 1) **¿Qué factores humanos han sido estudiados en su relación con la calidad del software?** Esta pregunta se responderá con la identificación y caracterización de los factores humanos que inciden en la calidad del software a partir de investigaciones primarias (objetivo específico 1).
- 2) **¿Qué factores humanos en la calidad del software han sido más estudiados y cuales falta por investigar?** Esta pregunta se responderá con la identificación y caracterización de los factores humanos que inciden en la calidad del software a partir de investigaciones primarias (objetivo específico 1).
- 3) **¿Qué buenas prácticas relacionadas con los factores humanos se identifican para posibilitar el desarrollo de software de calidad?** Esta pregunta se responderá con la identificación de las buenas prácticas, relacionadas con los factores humanos identificados, que posibilitan el desarrollo de software de calidad (objetivo específico 2).

- 4) **¿Qué parámetros o indicadores para evaluar los factores humanos se identifican y que podrían utilizarse en el control y aseguramiento de la calidad?** Esta pregunta se responderá con la identificación de algunos parámetros o indicadores para evaluar los factores humanos como parte de los procesos de control y aseguramiento de la calidad del software (objetivo específico 3).
- 5) **¿Cómo podrían clasificarse los factores humanos en la calidad del software?** Esta pregunta se responderá con una propuesta de clasificación de los factores humanos que inciden en la calidad del software (objetivo específico 4).
- 6) **¿Qué factores humanos que inciden en la calidad del software se reconocen en el contexto salvadoreño?** Esta pregunta se responderá con el reconocimiento de los factores humanos que según la perspectiva y experiencia de profesionales en el desarrollo de software en El Salvador inciden en la calidad del software (objetivo específico 5).
- 7) **¿Qué grado de importancia se le dan a los factores humanos en la calidad del software en el contexto salvadoreño?** Esta pregunta se responderá con la determinación del grado de importancia que profesionales en el desarrollo de software en El Salvador dan a los factores humanos en la calidad del software (objetivo específico 6).

FIGURA NO. 3 OBJETIVOS ESPECÍFICOS Y PREGUNTAS DE INVESTIGACIÓN



Nota. Elaboración propia.

2.4 Justificación

Aunque existe cierta tendencia a enfatizar los procedimientos, herramientas y sistemas como determinantes de la calidad del software, como afirma Gillies, la calidad del software es acerca de las personas:

“Son personas y las organizaciones conformadas por personas los que tienen los problemas a ser abordados por el software, son personas las que definen el problema y especifican la solución, son personas las que implementan el diseño y producen el código, son personas las que prueban el código, son personas las que usan el sistema final y las que harán juicios acerca de la calidad de toda la solución” [5, p. 15].

Se reconoce que los factores humanos son el origen de la mayor parte de los problemas en el desarrollo de software [13, p. 7] [14, pp. 74-76], y también se reconoce que si el empleado desarrollando el software no tiene los conocimientos y habilidades apropiadas, o no tiene el interés, la calidad del software se verá afectada negativamente [31, p. 47].

Además, las principales causas de la mala calidad del software tienen su origen en el factor humano: desconocimiento del negocio, desconocimiento de la tecnología, plazos irrealistas, mala ingeniería de software y malas decisiones de adquisición [35].

Por estos motivos en esta investigación se propone estudiar los factores humanos que inciden en la calidad del software.

También, en la investigación preliminar no se encontró respuesta a la pregunta de cuáles son los factores humanos que inciden en la calidad del software. Para el caso, Hemstra observa que dicha pregunta no es un pregunta fácil de responder porque el número de factores que inciden en la calidad del software es grande, la relevancia o impacto de un factor específico depende de las características de la organización y del entorno de desarrollo, y no hay un consenso acerca de cuáles factores deben ser considerados [31, p. 47].

Sin embargo, también se estableció que existen estudios que identifican individualmente algunos factores humanos y que confirman o no su incidencia en la calidad del software en contextos particulares. Por lo que, consideramos que la identificación de los factores humanos a partir del estudio de la literatura existente, la compilación de estudios e investigaciones publicados alrededor del tema y el reconocimiento de los mismos con profesionales en el desarrollo de software de El Salvador, es un aporte útil en el área.

La investigación tiene además de valor teórico, implicaciones profesionales y prácticas.

El identificar los factores humanos que inciden en la calidad del software, clasificarlos y compilar los estudios que confirman su incidencia proveerá de información para comprenderlos y reconocer las buenas prácticas que posibilitan el desarrollar software de calidad, y servir de apoyo en la evaluación de desarrolladores, equipos de desarrollo y el entorno organizacional como parte de los procesos de control y aseguramiento de la calidad del software.

2.5 Alcances

Los límites conceptuales y metodológicos de la investigación son:

- 1) Se identifican los factores humanos que inciden en la calidad del software que se originan en las competencias y habilidades del recurso humano involucrado en el desarrollo de software (usuarios o clientes, directores de proyectos, arquitectos de sistemas, analistas, desarrolladores, etc.), de sus relaciones con su entorno organizacional y de sus interacciones con otros individuos en el ambiente de trabajo.

- 2) Se consideran factores humanos relacionados con características individuales como la personalidad, el carácter y el estado de ánimo, que originalmente se había establecido que no se considerarían, pues las entrevistas establecieron la importancia de estos.
- 3) Se considera el factor humano relacionado con el entorno ambiental de trabajo, que originalmente se había establecido que no se consideraría, pues las entrevistas establecieron la importancia de éste.
- 4) No se consideran factores humanos relacionados con el entorno ambiental, el entorno social fuera del ambiente de trabajo, y con características individuales como la edad, sexo, raza y discapacidades.
- 5) Se identifican inicialmente los factores humanos que inciden en la calidad del software a partir de una revisión de la literatura.
- 6) Se identifican las buenas prácticas, y algunos parámetros e indicadores asociados solamente a los factores humanos identificados, y que según la revisión de la literatura posibilitan el desarrollar software de calidad.
- 7) Se determina cuáles de esos factores humanos son reconocidos por profesionales en El Salvador que trabajan o han trabajado en el desarrollo de software en empresas de cualquier tamaño, tanto del sector de la industria de software como de los sectores industrial, comercial o de servicio, ya sean privadas o públicas.
- 8) Se determina el grado de importancia que se le dan a los factores humanos en la calidad del software según la perspectiva y experiencia de los profesionales participantes en la investigación.

2.6 Limitaciones

Se distinguen las siguientes limitaciones:

- 1) La complejidad del tema de estudio. Como se dijo, el tema es complejo debido a la gran cantidad y variedad de factores humanos que pueden incidir en la calidad del software, a las interrelaciones y dependencias entre ellos [31, p. 47].
- 2) La imposibilidad de identificar todos los factores humanos que inciden en la calidad del software. La revisión de la literatura y la investigación de campo propuestas no aseguran que se identifique la totalidad de los factores humanos que pueden incidir en la calidad del software.
- 3) La confiabilidad y validez de la investigación de campo como medio de validación de los factores humanos identificados. Por tratarse de una investigación con características cualitativas y por tanto, con un componente subjetivo muy importante, su validez y confiabilidad se limita al contexto en el que fue validada dejando abierta la posibilidad a otros estudios que puedan confirmar su confiabilidad y validez al llegar a conclusiones equivalentes en otros contextos.
- 4) La imposibilidad de generalizar los hallazgos de la investigación de campo. El reconocimiento de los factores humanos que inciden en la calidad del software con los profesionales participantes, por su alcance, limita su aplicabilidad y la generalización de los hallazgos en otros contextos.

2.7 Enfoque de la investigación

Pese a que las ideas filosóficas no son evidentes en una investigación, influyen la práctica y deben ser identificadas [43, p. 5]. Dada la complejidad del tema se ha optado por adherirse a las ideas de una visión filosófica **constructivista**.

El constructivismo sostiene que los individuos buscan entender el mundo en el que viven y trabajan, desarrollando significados subjetivos de sus experiencias. Estos significados son variados y múltiples, llevando al investigador a buscar la complejidad de esos puntos de vista en lugar de estrechar el

significado en unas pocas categorías e ideas. Bajo este enfoque se confía tanto como sea posible en los puntos de vista de los participantes acerca de la situación estudiada [43, p. 8].

Los supuestos de esta visión son [43, p. 8]:

- 1) **El significado es construido por los seres humanos a medida que se involucran con el mundo que están interpretando.** El investigador tiende a usar preguntas abiertas y generales permitiendo al participante construir el significado de la situación, típicamente en discusiones o interacciones cara a cara. El objetivo del investigador es encontrar sentido o interpretar los significados tienen acerca del problema.
- 2) **Los seres humanos se involucran en el mundo y encuentran su significado basados en sus perspectivas históricas y culturales.** El investigador busca entender el contexto situándose en el ambiente natural de los participantes (donde enfrentan el problema de estudio), visitándolos y recolectando sus opiniones y experiencias personalmente. También el investigador debe reconocer que sus propios conocimientos y experiencias dan forma a su interpretación.
- 3) **La generación de conocimiento es social, surgiendo de la interacción del individuo con su comunidad.** El investigador genera conocimiento a partir de información recolectada en el campo con los participantes, trabajando de lo particular a lo general mediante una técnica de análisis inductivo.

2.8 Estrategias de investigación

El diseño de la investigación comprende dos estrategias generales: la **revisión de la literatura** y la **investigación de campo**.

La **revisión de la literatura** tiene por objetivo enmarcar el problema y contextualizar el proceso de investigación proveyendo el punto de partida para la **identificación y caracterización de los factores humanos en la calidad del software** así como las buenas prácticas que posibilitan el desarrollo de software de calidad, y el desarrollo de una clasificación de los mismos basados en investigaciones primarias relevantes al tema y a cada una de las preguntas de investigación.

La revisión de la literatura es el insumo para la segunda parte de la investigación, la **investigación de campo**, la cual está dirigida a profesionales en el desarrollo de software de El Salvador. La investigación de campo comprende dos fases: la investigación cualitativa y la investigación cuantitativa. La investigación de campo tiene por objetivo **contextualizar y reconocer los factores humanos** identificados en la revisión de la literatura con profesionales que trabajen o hayan trabajado en el desarrollo de software en El Salvador. Establecerá los factores humanos que desde la perspectiva de dichos profesionales inciden en la calidad del software y el grado de relevancia que le dan a cada uno de ellos.

2.9 Tipo de investigación

Se desarrollará una investigación **mixta y secuencial**. **Mixta** porque incorpora elementos de una **investigación cualitativa** con el objetivo de entender el problema de estudio; y también de una **investigación cuantitativa** con el objetivo de elaborar o expandir los hallazgos. Y **secuencial** porque no se realizarán ambas investigaciones en paralelo, sino que, una después de la otra.

El objetivo de contemplar los dos tipos de investigación reside en la conveniencia de tener varias fuentes de información y métodos de recolección de datos, y posibilitar la triangulación de datos con el objetivo de validarlos [44, p. 418].

2.9.1 Fase 1: Investigación cualitativa

La investigación es **cualitativa** en su primera fase, con el objetivo de entender el problema basándose en las opiniones y experiencias de cada uno de los participantes. El foco de la investigación se centra en el significado e interpretación que cada individuo de un grupo pequeño de participantes da al problema, para iniciar con el reconocimiento de los factores humanos que se consideran relevantes

La estrategia de investigación a utilizar en esta fase es la de una **investigación fenomenológica**. La investigación fenomenológica es una estrategia en la cual el investigador identifica la esencia de las experiencias humanas acerca del fenómeno según es descrito por los participantes [43, p. 15].

En cuanto al número de participantes en investigaciones fenomenológicas, Creswell menciona que el número de participantes puede ser entre 5 y 25; y Morse menciona que deben ser al menos 6 participantes [45].

2.9.2 Fase 2: Investigación cuantitativa

Es **cuantitativa** en una segunda fase, con el objetivo de ampliar el entendimiento del problema con un grupo de participantes más grande llegando a reconocer los factores humanos que según la perspectiva y experiencia de los profesionales participantes inciden en la calidad del software y el grado de importancia que les dan.

La estrategia de investigación a utilizar en esta fase es la de una **investigación descriptiva y no experimental**. Descriptiva, pues su objetivo es describir el fenómeno y sus componentes [44, p. 89]; y **no experimental** porque no busca establecer si una situación produce un resultado específico sino que busca proveer una descripción numérica de tendencias, actitudes u opiniones de los participantes del estudio [43, p. 12].

2.10 Participantes de la investigación

En ambas fases de la investigación participarán **profesionales que trabajan o han trabajado en el desarrollo software** en El Salvador.

2.10.1 Tipo de muestreo

En ambas fases de la investigación el muestreo es **no probabilístico de expertos**, pues no se busca generalizar los hallazgos en otros contextos. Los profesionales (casos o unidades) fueron seleccionados según el propósito de la investigación, y sin pretender que los casos fueran representativos de la población.

2.10.2 Caracterización de la muestra

Las características de los participantes en la investigación son:

- 1) **Área profesional**, la investigación está orientada a profesionales con experiencia en el área del desarrollo de software. Se entenderá esta área profesional como la participación en el proceso de desarrollo de software o proceso de software, el cual comprende una secuencia de actividades que conducen a la elaboración de un producto de software, como [12, p. 28]:
 - a) **Especificación del software**, donde el cliente y los responsables de desarrollarlo definen los requisitos y diseñan el software que se desarrollará y las restricciones en su operación.
 - b) **Desarrollo del software**, donde se diseña y programa el software.
 - c) **Validación o pruebas del software**, donde se verifica el software para asegurar que sea lo que el cliente requiere.
 - d) **Evolución o mantenimiento del software**, donde se modifica el software para reflejar los requisitos cambiantes del negocio que el software soporta.

- 2) **Situación laboral**, los profesionales participantes en el estudio podrán estar activos o inactivos laboralmente, esto es, que trabajen o hayan trabajado en el desarrollo de software.
- 3) **Ubicación geográfica**, los profesionales participantes deberán trabajar o haber trabajado en El Salvador.
- 4) **Tipo de empresa**, los profesionales participantes deberán trabajar o haber trabajado en empresas de cualquier tamaño, tanto del sector de la industria de software como de los sectores industrial, comercial o de servicio, ya sean privadas o públicas de El Salvador.
- 5) **Nivel funcional**, los profesionales participantes deberán trabajar o haber trabajado en cualquiera de las siguientes áreas funcionales:
 - a) **Nivel estratégico**, desempeñando cargos como directores, gerentes o jefes de tecnologías de información o informática.
 - b) **Nivel de gestión**, desempeñando cargos como directores, administradores o gestores de proyectos de desarrollo, etc.
 - c) **Nivel operativo**, desempeñando cargos como analistas, diseñadores, arquitectos, desarrolladores, probadores de software, etc.
- 6) **Nivel de estudios**, el grado académico requerido para cada fase.
- 7) **Años de experiencia**, el número mínimo de años de experiencia requerido para cada fase.
- 8) **Características biográficas**, se refieren a la nacionalidad, sexo, edad, raza, religión, discapacidades, etc. del participante. Para ambas fases del estudio las características biográficas no son relevantes.

2.10.3 Criterios de selección de selección de la muestra

Los criterios de selección de los participantes en la investigación son:

TABLA NO. 7 CRITERIOS DE SELECCIÓN DE LA MUESTRA DE EXPERTOS

Criterios	Investigación cualitativa	Investigación cuantitativa
Área profesional	Desarrollo de software.	Desarrollo de software.
Ubicación geográfica	El Salvador.	El Salvador.
Nivel de estudios	Licenciatura o ingeniería si no tuviera la experiencia.	Cualquier nivel.
Años de experiencia	Mínimo 10 años si no tuviera el nivel de estudios indicado arriba o 5 años si lo tuviera.	Cualquiera.

Nota. Elaboración propia.

Originalmente se pensó, en la fase cuantitativa excluir participantes con menos de 3 años de experiencia. Sin embargo a partir de la revisión de la literatura, que indica que los años de experiencia no es una característica que se considere tenga impacto en la calidad del software, y a partir de los resultados en las entrevistas, se concluyó que no era prudente pues se excluiría a profesionales recién graduados que inician su carrera, y que tienen desde ya cierta experiencia, y podrían aportar su visión a la investigación.

2.10.4 Tamaño de la muestra

Se tomará en consideración los tamaños recomendados para la estrategia de investigación seleccionada (fenomenológica), que según Creswell pueden ser entre 5 y 25 participantes; y según Morse debe ser al menos de 6 participantes [45].

Así para la fase de la **investigación cualitativa se seleccionaron 5 participantes**, y para la fase de la **investigación cuantitativa se seleccionaron 40 participantes** según las características detalladas en la tabla anterior.

2.10.5 Selección de la muestra

Para la selección de los participantes se utiliza un muestreo de expertos en cadena o por redes [44, pp. 388-390].

2.11 Recolección de datos

El investigador es el principal instrumento en la recolección de datos, y se auxilia de diversas fuentes de datos y técnicas de recolección, según se detalla a continuación.

2.11.1 Fuentes de datos

La investigación considera la utilización de los tipos de fuentes de datos siguientes [46, pp. 62-65]:

- 1) **Fuentes primarias:** son los documentos, investigaciones o información generada por los participantes en el período en que se realiza el estudio a partir de entrevistas y cuestionarios.
- 2) **Fuentes secundarias:** son los reportes que resumen documentos, investigaciones o información de fuentes primarias.
- 3) **Fuentes terciarias:** son libros o artículos que resumen de fuentes secundarias.

Con ello se llega a identificar a través de más de una fuente un conjunto de factores humanos que inciden en la calidad del software, permitiendo su triangulación y garantizando la confiabilidad y validez del estudio.

2.11.2 Técnicas de recolección de datos

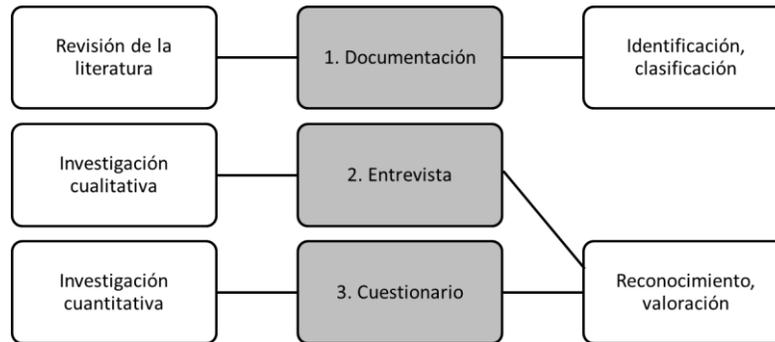
El tema de investigación es complejo debido a la gran cantidad y variedad de factores humanos que pueden incidir en la calidad del software, a las interrelaciones entre ellos y con otros factores no humanos, a su dependencia del entorno organizacional y de trabajo, y al hecho que aún no hay un consenso sobre cuáles deben ser considerados [31, p. 47], y una identificación completa de los mismos es hasta cierto punto imposible.

Por ello su identificación se hace inicialmente a partir de la **revisión de la literatura**. Luego, se desarrolla una **investigación de campo** para reconocer los factores identificados en la revisión de la literatura en el contexto salvadoreño. En la investigación de campo se utilizan técnicas **cualitativas** y **cuantitativas** de recolección de datos. Iniciando con técnicas cualitativas que se ajusten al fenómeno o situación a investigar se busca entender el problema; y luego, con técnicas cuantitativas se busca elaborar o expandir los hallazgos con una muestra mayor.

Se propone la utilización de ambos métodos de recolección de datos, cualitativos y cuantitativos con el objetivo de identificar a través de más de un método de recolección un conjunto de factores humanos que se reconoce inciden en la calidad del software permitiendo su triangulación y garantizando la confiabilidad y validez del estudio.

Así, las técnicas de recolección de datos son:

FIGURA NO. 4 TÉCNICAS DE RECOLECCIÓN DE DATOS



Nota. Elaboración propia.

2.11.2.1 DOCUMENTACIÓN

En toda investigación es necesario estudiar la literatura acerca del tema estudio [43, p. 57]. La revisión de la literatura cumple varios propósitos: provee al investigador de resultados de otros estudios que están relacionados con el tema de la investigación, enmarca la investigación en los temas más generales que la rodean, y provee de un marco de trabajo para establecer la importancia de la investigación. [43, p. 60].

La revisión de la literatura es el “*paso de investigación que consiste en detectar, consultar y obtener la bibliografía y otros materiales útiles para los propósitos del estudio, de los cuales se extrae y recopila información relevante y necesaria para el problema de investigación*” [44, p. 61].

TABLA NO. 8 USO DE LA LITERATURA EN INVESTIGACIONES

Uso de la literatura	Tipos de investigación
1) Para enmarcar el problema al inicio de la investigación.	En todas las investigaciones cualitativas (investigaciones narrativas, fenomenológicas, de teoría fundamentada, etnografías, casos de estudio).
2) Para presentar lo que se ha estudiado a la fecha alrededor del problema, como una revisión de la literatura o estado del arte.	En investigaciones basadas fuertemente en una teoría, como etnografías o de teoría fundamentada
3) Para comparar y contrastar los hallazgos de la investigación.	En todos los tipos de investigaciones cualitativas (investigaciones narrativas, fenomenológicas, de teoría fundamentada, etnografías, casos de estudio), pero principalmente en los de teoría fundamentada

Nota. Adaptado de *Research Design* [43, p. 62].

Algunas estrategias para la revisión de la literatura son [43, p. 64]:

- 1) Iniciar con la identificación de palabras claves.
- 2) Buscar bibliografía y artículos de investigación relacionados con esas palabras claves.
- 3) Revisar la bibliografía y artículos para establecer si contribuirán a la comprensión del problema.
- 4) Desarrollar un mapa de la literatura (una imagen visual de cómo contribuye al tema de la investigación).
- 5) Hacer resúmenes de la literatura relevante.
- 6) Elaborar la revisión de la literatura, estructurando temáticamente y organizando los conceptos importantes.

En esta investigación, el uso que se le da a la revisión de la literatura es el de **enmarcar el problema al inicio de la investigación** para proveer un mayor entendimiento del problema y además permitir identificar los factores humanos en la calidad del software así como las buenas prácticas que posibilitan el desarrollo de software de calidad y proponer una clasificación inicial que es validada en

la segunda etapa de la investigación. Se estudian las opiniones de expertos en libros, resultados de estudios e investigaciones publicadas, etc. de fuentes primarias, secundarias o terciarias.

2.11.2.2 ENTREVISTA

La técnica de la entrevista se utiliza en la segunda parte de la investigación, la investigación de campo, en su fase cualitativa con el objetivo de validar en el contexto salvadoreño los factores humanos en la calidad del software obtenidos a partir de la revisión de literatura.

En investigaciones cualitativas el investigador conduce entrevistas cara a cara con los participantes, entrevistas telefónicas, o entrevistas con grupos focales de 6 a 8 entrevistados en cada grupo [43, p. 240]. Estas entrevistas comprenden preguntas no estructuradas o semiestructuradas y generalmente abiertas que son pocas en número [43, p. 240].

TABLA NO. 9 VENTAJAS Y DESVENTAJAS DE LA TÉCNICA DE LA ENTREVISTA

Ventajas	Desventajas
<ul style="list-style-type: none"> - Útil cuando los participantes no pueden ser observados directamente. - Los participantes pueden proveer información histórica o información basada en su experiencia pasada. - El entrevistador puede controlar la secuencia y la intención de las preguntas. 	<ul style="list-style-type: none"> - Provee información indirecta filtrada a través de los puntos de vista de los entrevistados. - Provee información en un lugar designado en vez del lugar natural en que realiza o desarrolla el fenómeno o situación investigada. - La presencia del investigador puede influenciar la respuesta. - No todas las personas son igualmente perceptivas y elocuentes.

Nota. Adaptado de Research Design [43, p. 241].

Se ha elegido la técnica de la entrevista porque, dado el diseño del estudio el fenómeno a investigar no puede ser estudiado mediante la observación directa, que es otra de las técnicas que usualmente se utilizan en investigaciones cualitativas.

Considerando estos lineamientos, en la investigación se utilizan **entrevistas cara a cara, y uno a uno con los participantes**. Las entrevistas comprenden un **pequeño número de preguntas semiestructuradas y abiertas**.

También, se consideran las siguientes estrategias [43, p. 242]:

- 1) Planear y desarrollar un protocolo de la entrevista que contenga el detalle de las instrucciones y los procedimientos a realizar durante la entrevista.
- 2) Desarrollar un conjunto de preguntas para guiar la entrevista.
- 3) Diseñar un registro para la entrevista en el cual no solo se registre las respuestas sino también los datos que caracterizan a cada participante (fecha, lugar, entrevistado, entrevistador).
- 4) Tomar nota durante las entrevista.
- 5) Grabar la entrevista, si es permitido por el participante, y luego transcribirla.

2.11.2.3 CUESTIONARIO

La técnica del cuestionario se utiliza en la segunda parte de la investigación, la investigación de campo en su fase cuantitativa, con el objetivo de reconocer y determinar el grado de importancia que se les asigna a los factores humanos en la calidad del software.

El cuestionario es el instrumento más utilizado para recolectar datos. Consiste en un conjunto de preguntas respecto a una o más variables a medir, y debe ser congruente con el planteamiento del problema y los objetivos de la investigación [44, p. 217]. El cuestionario permite obtener descripciones cuantitativas o numéricas de tendencias, actitudes u opiniones de una población estudiando una muestra de dicha población [43, p. 200].

Los cuestionarios pueden tener dos tipos de preguntas [44, pp. 217-220]:

- 1) **Preguntas cerradas:** contienen categorías u opciones de respuesta que han sido previamente delimitadas. Es decir se presentan las posibilidades de respuesta a los participantes, quienes deben limitarse a estas. Las respuestas pueden presentar dos o más posibilidades. El participante, dependiendo del diseño del cuestionario podrá seleccionar una respuesta o más de una respuesta. También, se le puede pedir al participante jerarquizar opciones, o asignar puntaje a diversas cuestiones en base a un rango determinado o puntajes discretos.
- 2) **Preguntas abiertas:** no delimitan de antemano las alternativas de respuesta, por lo cual el número de alternativas de respuesta es muy elevado.

Algunas estrategias para el diseño de las preguntas del cuestionario son [44, p. 224]:

- 1) Ser claras, precisas y comprensibles por los participantes de la investigación. Deben evitarse términos confusos, ambiguos y de doble sentido.
- 2) Deben ser breves. Sin embargo, no se debe sacrificar la claridad por la concisión.
- 3) Deben formularse con un vocabulario simple, directo y familiar para los participantes.
- 4) No deben incomodar a la persona encuestada ni ser percibidas como amenazantes y nunca ésta debe sentir que se le enjuicia.
- 5) Deben referirse a un solo aspecto o relación lógica.
- 6) No deben inducir las respuestas.
- 7) No deben negar el asunto que se interroga.
- 8) No deben ser discriminatorias (racistas, sexistas, etc.).

Dado que en la fase cuantitativa se busca reconocer los factores humanos identificados y determinar el grado de importancia que se les asigna, el cuestionario es apropiado para obtener esa descripción numérica que se busca. Y considerando los lineamientos indicados y el objetivo que persigue el cuestionario es de **preguntas cerradas**.

2.12 Análisis y organización de datos

En investigaciones cuantitativas primero se recolectan los datos y luego se analizan, mientras que en investigaciones cualitativas, la recolección y el análisis ocurren en paralelo [44, p. 418].

En la investigación cualitativa se recibe de las entrevistas datos no estructurados, a los que se les da estructura. Estos datos son en esencia de dos tipos: observaciones del investigador y narraciones de los participantes. El proceso de análisis, ocurre en paralelo al de recolección, y en términos generales es el siguiente [44, p. 418]:

- 1) Explorar los datos.
- 2) Organizar los datos.
- 3) Describir las experiencias de los participantes.
- 4) Descubrir los conceptos, categorías, temas, patrones y sus vínculos.
- 5) Comprender el contexto que rodean los datos.
- 6) Reconstruir hechos e historias.
- 7) Vincular los resultados con el conocimiento disponible.
- 8) Generar las conclusiones.

Cuando ya no se encuentra información nueva (saturación) el proceso de análisis concluye. Pero si se descubren inconsistencias o falta de claridad en el entendimiento del problema planteado se regresa al campo para recolectar más datos [44, p. 419].

Para el análisis de datos cualitativos se utilizarán métodos de codificación abierta o de primer nivel con el fin de comparar unidades y crear categorías; y métodos de codificación axial o de segundo nivel para comparar categorías [44, p. 271].

En la investigación cuantitativa se reciben datos a partir de los cuestionarios que son analizados mediante métodos de análisis descriptivos, como: distribuciones de frecuencias, medidas de tendencia central, medidas de variabilidad y gráficas [44, p. 395].

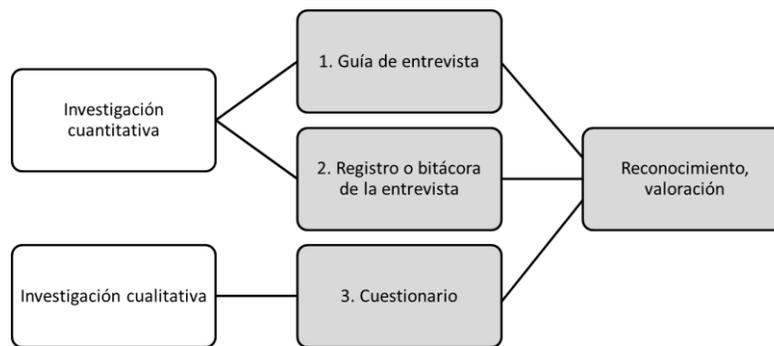
2.13 Instrumentos de investigación

Se presenta a continuación los instrumentos de recolección y organización de datos así como su diseño.

2.13.1 Instrumentos para la recolección de datos

Congruente con las técnicas de recolección de datos los instrumentos utilizados son:

FIGURA NO. 5 INSTRUMENTOS DE RECOLECCIÓN DE DATOS



Nota. Elaboración propia.

2.13.1.1 GUÍA DE ENTREVISTA

En la primera fase de la investigación de campo (investigación cualitativa) se utiliza la técnica de la **entrevista**. Con ella se busca reconocer los factores humanos que fueron identificados a partir de la revisión de literatura.

Las entrevistas son **cara a cara**, basándose en un **número pequeño de preguntas semiestructuradas y abiertas**. La guía de la entrevista comprende:

- 1) Las **preguntas**,
- 2) El **protocolo** de la entrevista, y
- 3) El **registro** o bitácora de la entrevista.

2.13.1.1.1 PREGUNTAS DE LA ENTREVISTA

Las preguntas de la entrevista han sido determinadas por los resultados de la revisión de la literatura y buscan satisfacer los objetivos específicos 5 y 6.

TABLA NO. 10 PREGUNTAS DE LA ENTREVISTA

No.	Pregunta principal	Preguntas secundarias
1	En su experiencia, ¿qué factores ha identificado inciden o contribuyen a la buena o mala calidad del software que se desarrolla?	¿Por qué considera que inciden? ¿Podría compartir situaciones o experiencias que ilustren porque considera que esos factores inciden en la calidad del software?
2	¿Cuáles de esos factores considera tienen origen humano?	
3	¿Cuáles de esos factores son los 3 que considera tienen mayor grado de incidencia o impacto en la calidad del software?	¿Por qué le da ese grado de importancia?

Nota. Elaboración propia.

Véase en el anexo No. 3 la guía de la entrevista.

2.13.1.1.2 PROTOCOLO DE ENTREVISTA

Se ha diseñado un protocolo de entrevista con el objetivo de guiar al entrevistador y posibilitar que el proceso de la entrevista sea repetible de participante a participante según un conjunto de lineamientos preestablecidos:

- 1) **Introducción:** en la introducción el entrevistador se presenta, da una descripción general de la investigación (propósito, participantes elegidos, motivo por el cual fueron seleccionados, utilización de los datos, etc.) e indica las características de la entrevista (confidencialidad, si se le puede grabar o no, duración aproximada, etc.)
- 2) **Establecimiento de la relación:** para establecer una buena relación inicial se pregunta sobre la información biográfica del entrevistado, teniendo cuidado de no incomodar, como: años de experiencia, edad, historial laboral, puesto actual, nivel educativo, etc. [43, p. 244].
- 3) **Establecimiento del contexto:** el entrevistador establece el contexto del entrevistado indagando sobre los conceptos y temas claves de la investigación (concepto de calidad, si evalúa la calidad del software que desarrolla, qué aspectos evalúa, qué métodos o técnicas utiliza para la evaluación, etc.)
- 4) **Realizar la entrevista:** el entrevistador realiza cada pregunta principal seguida de las preguntas secundarias que se detallan en la guía de entrevista según lo estime conveniente. Deberá solicitar a los participantes explicar sus ideas en detalle, o elaborar sobre lo que han dicho [43, p. 244]. También deberá registrar las respuestas en el registro o bitácora de la entrevista.
- 5) **Cierre:** el entrevistador agradece y reconoce el tiempo que el entrevistado ha compartido.

2.13.1.2 REGISTRO O BITÁCORA DE ENTREVISTA

Se ha diseñado una bitácora de la entrevista con el objetivo posibilitar un registro consistente de las respuestas de los participantes. La bitácora contiene información siguiente:

- 1) Fecha de la entrevista.
- 2) Hora de inicio y de fin de la entrevista.
- 3) Lugar (ciudad).
- 4) Datos biográficos del entrevistado (sexo, puesto, nivel educativo, puesto, experiencia).
- 5) Anotaciones (respuestas a cada una de las preguntas).

Véase en el anexo No. 4 la bitácora de la entrevista.

2.13.1.3 CUESTIONARIO

En la segunda fase de la investigación de campo (investigación cuantitativa) se utiliza la técnica del **cuestionario**. Con el cuestionario se busca validar los resultados de las entrevistas con un grupo de participantes más amplio, y además conducir a la valoración de los factores humanos.

El cuestionario es de **preguntas cerradas**. El diseño del cuestionario se ha hecho en base a la información producida por la investigación de campo en la fase cualitativa (entrevistas), y comprende:

- 1) Las **preguntas** a realizar, y
- 2) El **protocolo** a seguir para administrar el cuestionario.

2.13.1.3.1 PREGUNTAS DEL CUESTIONARIO

Las preguntas del cuestionario han sido determinadas por los resultados de la revisión de la literatura y la primera fase de la investigación de campo, y buscan satisfacer los objetivos específicos 5 y 6.

TABLA NO. 11 PREGUNTAS DEL CUESTIONARIO

No.	Pregunta
Factores humanos relacionados con el individuo	
1	¿Inciden las competencias en la calidad del software?
2	¿Qué competencias considera inciden en la calidad del software?
3	¿Cómo incide la experiencia en la calidad del software?
4	¿Cómo incide el profesionalismo en la calidad del software?
5	¿Cómo incide el empoderamiento en la calidad del software?
6	¿Cómo incide la conciencia de calidad en la calidad del software?
7	¿Cómo incide la creatividad en la calidad del software?
8	¿Incide la personalidad en la calidad del software?
9	¿Qué rasgos de personalidad considera inciden en la calidad del software?
10	¿Incide el carácter en la calidad del software?
11	¿Qué rasgos del carácter considera inciden en la calidad del software?
12	¿Incide el estado de ánimo en la calidad del software?
Factores humanos relacionados con la organización	
13	¿Incide la estructura organizativa en la calidad del software?
14	¿Qué aspectos de la estructura organizativa considera inciden en la calidad del software?
15	¿Incide la cultura organizacional en la calidad del software?
16	¿Qué aspectos de la cultura organizacional considera inciden en la calidad del software?
17	¿Incide el estilo de liderazgo en la calidad del software?
18	¿Qué estilos de liderazgo considera inciden en la calidad del software?
19	¿Cómo incide la gestión del conocimiento en la calidad del software?
20	¿Cómo incide el tamaño del equipo de desarrollo en la calidad del software?
21	¿Qué tamaño de equipo de desarrollo considera óptimo?
22	¿Incide la distribución del equipo de desarrollo en la calidad del software?
23	¿Qué tipo de distribución del equipo considera incide en la calidad del software?
24	¿Cómo incide la cohesión del equipo de desarrollo en la calidad del software?
Factores humanos relacionados con el involucramiento en el trabajo	
25	¿Cómo incide la motivación en la calidad del software?
26	¿Incide la satisfacción en la calidad del software?
27	¿Qué aspectos relacionados con la satisfacción considera inciden en la calidad del software?
28	¿Cómo incide el sentido de pertenencia del recurso humano en la calidad del software?
29	¿Incide la propiedad del código en la calidad del software?
30	¿Qué modelos de propiedad del código inciden en la calidad del software?
31	¿Incide el entorno ambiental en la calidad del software?
32	¿Qué aspectos del entorno ambiental considera inciden en la calidad del software?
33	¿Qué otros factores humanos, que no se hayan mencionado, considera inciden en la calidad del software?
Valoración de los factores humanos	
34	¿Qué factores humanos considera inciden más en la calidad del software?

Nota. Elaboración propia.

Véase en el anexo No. 5 el diseño del cuestionario.

2.13.1.3.2 PROTOCOLO DEL CUESTIONARIO

El cuestionario es **auto administrado y en línea**, y el protocolo está descrito en el párrafo introductorio del mismo, y comprende:

- 1) **Descripción de la investigación:** descripción general de la investigación y las indicaciones generales para completar el cuestionario.
- 2) **Identificación del entrevistado:** se pregunta sobre la información biográfica del entrevistado, como: años de experiencia, edad, historial laboral, puesto actual, nivel educativo, etc.
- 3) **Establecimiento del contexto:** se pregunta sobre el contexto del entrevistado indagando sobre los conceptos o temas claves de la investigación (si evalúa la calidad del software que desarrolla, qué aspectos evalúa, qué métodos o técnicas utiliza para la evaluación, etc.)
- 4) **Completar el cuestionario:** el participante completa el cuestionario.
- 5) **Cierre:** se agradece y reconoce el tiempo que el participante ha compartido.

2.13.2 Instrumentos para la organización de datos

Los instrumentos de organización de datos que se utilizan son:

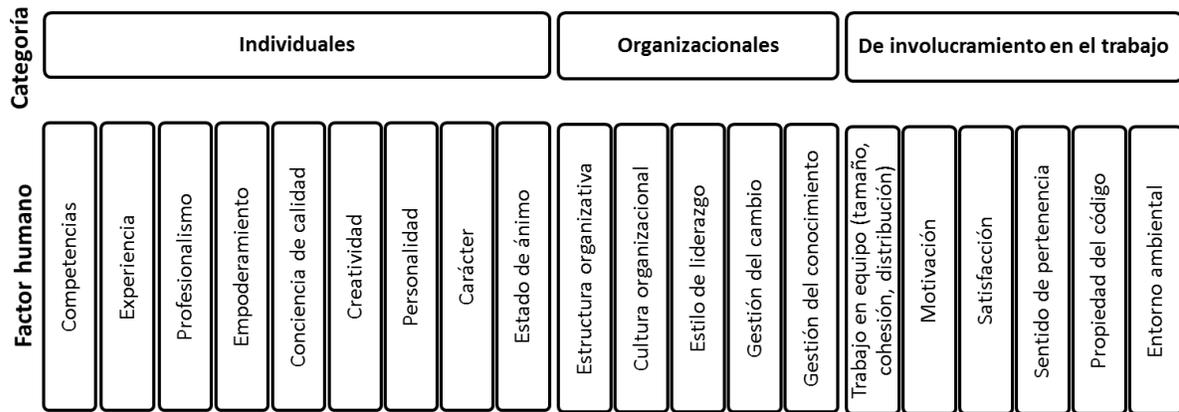
2.13.2.1 SISTEMA DE CLASIFICACIÓN

Un sistema de clasificación de los factores humanos que inciden en la calidad del software según dos niveles:

- 1) **Categoría de factor humano**, las categorías identificadas en la revisión de la literatura.
- 2) **Factor humano**, los factores humanos identificados en la revisión de la literatura.

A continuación se presenta el sistema de clasificación:

FIGURA NO. 6 SISTEMA DE CLASIFICACIÓN



Nota. Elaboración propia.

2.13.2.2 SISTEMA DE VALORACIÓN

La valoración se realiza para **cada factor humano**, indicando si se considera que el factor incide o no en la calidad del software. Para algunos de ellos, se pide que se indique si la incidencia se considera positiva o negativa.

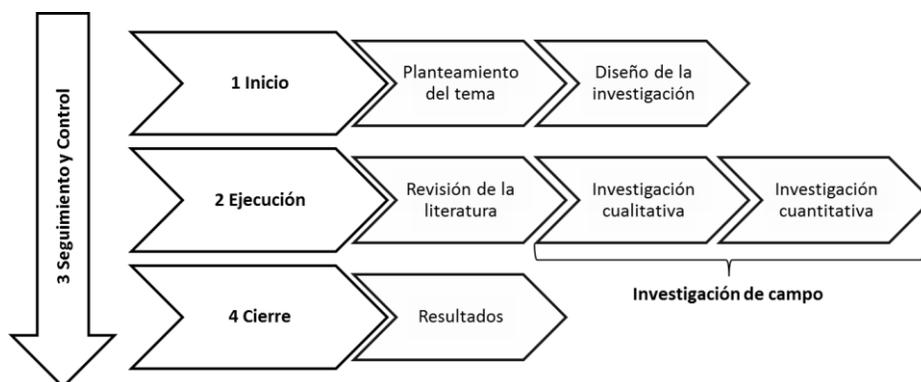
También se pide valorarlos indicando cuales son los 5 factores humanos que considera más relevantes.

2.14 Proceso de investigación

El proceso de investigación es emergente y abierto, y comprende las siguientes etapas:

- 1) Inicio.
- 2) Ejecución.
- 3) Seguimiento y control.
- 4) Cierre.

FIGURA NO. 7 PROCESO DE INVESTIGACION



Nota. Elaboración propia.

2.14.1 Inicio

El propósito de esta etapa es desarrollar el **planteamiento del problema y el diseño de la investigación**; y comprende las actividades desde la concepción hasta la definición del tema y el desarrollo de la metodología.

El entregable de esta etapa es el “**Perfil del trabajo de graduación**” o “**Anteproyecto**”.

2.14.2 Ejecución

Comprende las actividades para el desarrollo del proyecto de investigación.

- 1) La revisión de la literatura, y
- 2) La investigación de campo.

2.14.2.1 REVISIÓN DE LA LITERATURA

La revisión de la literatura tiene por propósito satisfacer los objetivos específicos 1 al 4 que competen a la **identificación y caracterización de los factores humanos** que según la revisión de la literatura inciden en la calidad del software, las buenas prácticas en esos factores humanos que posibilitan el desarrollo de software de calidad, los parámetros o indicadores y la clasificación.

El entregable es el documento de “**Revisión de la literatura**”.

2.14.2.2 INVESTIGACIÓN DE CAMPO

La investigación de campo tiene por propósito satisfacer los objetivos específicos 5 y 6 que competen al **reconocimiento de los factores humanos en la calidad del software y su grado de importancia en el contexto salvadoreño**.

La investigación de campo comprende las siguientes fases:

- 1) Fase 1: Investigación cualitativa.

2) Fase 2: Investigación cuantitativa.

Las actividades que comprende van desde la selección de los participantes, la ejecución de la fase de investigación cualitativa, la ejecución de la fase de investigación cuantitativa, la organización y el análisis y discusión de los resultados.

El entregable es el documento de “**Resultados de la investigación de campo**”.

2.14.3 Seguimiento y control

Comprende las actividades de seguimiento del proyecto de investigación.

2.14.4 Cierre

Comprende las actividades de presentación de los informes finales de la investigación, la “**Tesina**” y el “**Artículo**”.

2.15 Entregables

Los entregables de la investigación son:

2.15.1 Perfil del trabajo de graduación

El documento esboza y sintetiza el planteamiento del problema y el diseño de la investigación. Establece en que consiste la investigación, su contexto, por qué la investigación es relevante, cuáles son sus límites conceptuales y metodológicos, la propuesta metodológica, las actividades necesarias, y los productos resultantes.

El contenido de este documento es:

- 1) Portada
- 2) Índice
- 3) Resumen
- 4) Descripción
- 5) Preguntas de investigación
- 6) Alcances y limitaciones
- 7) Metodología
- 8) Entregables
- 9) Calendario de actividades
- 10) Recursos y presupuesto
- 11) Bibliografía

2.15.2 Revisión de la literatura

Este documento es el resultado de la revisión de la literatura. Identifica y caracteriza los factores humanos que según la literatura estudiada inciden en la calidad del software, las buenas prácticas que posibilitan el desarrollo de software de calidad, los parámetros o indicadores y la propuesta de clasificación.

El contenido de este documento es:

- 1) Portada
- 2) Índice
- 3) Introducción
- 4) Factores humanos en la calidad del software
- 5) Clasificación de los factores humanos

- 6) Buenas prácticas de los factores humanos
- 7) Indicadores para evaluación de los factores humanos
- 8) Bibliografía

2.15.3 Resultados de la investigación de campo

Este documento es el resultado de la investigación cualitativa y cuantitativa para contextualizar el problema en El Salvador.

El contenido de este documento es:

- 1) Portada
- 2) Índice
- 3) Introducción
- 4) Fase 1: Investigación cualitativa
- 5) Fase 2: Investigación cuantitativa
- 6) Bibliografía

2.15.4 Tesina

El entregable “**Tesina: Factores humanos de la calidad del software**” es el informe final y tiene por objetivo compilar los entregables anteriores y comunicar los resultados de la investigación.

El contenido de este entregable es:

- 1) Portada
- 2) Índice
- 3) Introducción
- 4) Marco conceptual
- 5) Marco metodológico
- 6) Revisión de la literatura
- 7) Investigación de campo
- 8) Conclusiones
- 9) Bibliografía

2.15.5 Artículo

El entregable “**Artículo: Factores humanos de la calidad del software**” es un resumen de la tesina con fines de publicación y el requisito final del trabajo de graduación.

El contenido de este entregable es:

- 1) Título
- 2) Resumen
- 3) Introducción
- 4) Antecedentes
- 5) Metodología
- 6) Revisión de la literatura
- 7) Resultados
- 8) Discusión
- 9) Conclusiones
- 10) Bibliografía

3 Revisión de la literatura

En el desarrollo de software participan tanto elementos tecnológicos como humanos. Sin embargo, pese a esta doble naturaleza, a menudo, cuando se habla de calidad del software, casi siempre, se pone más atención a aquellos elementos y actividades de carácter tecnológico, y se pone muy poca a aquellas de carácter humano.

DeMarco y Lister en su libro *PeopleWare* afirman que “*la mayor parte de los problemas de nuestro trabajo no son tanto tecnológicos sino que sociológicos en naturaleza*” [13, p. 4] y entre los problemas que más se mencionan están: la dificultad en la comunicación, el no contar con recurso humano idóneo, la alta rotación de personal, el desencanto con los jefes o clientes, y la falta de motivación [13, p. 4].

Los factores humanos, sin duda, inciden en la calidad del software y cada día son reconocidos como los más importantes. La falta de motivación y compromiso con la organización, los problemas de comunicación, los empleados insatisfechos, los equipos no integrados, un mal liderazgo inciden, todos, en la calidad del software.

Pero, ¿qué factores humanos inciden en la calidad del software? Según Hemstra [31, p. 47] esta no es un pregunta fácil de responder porque el número de factores que podrían incidir en la calidad es grande, la relevancia o impacto de un factor específico depende de las características de la organización y del entorno de desarrollo, y no hay un consenso acerca de cuáles factores deben ser considerados.

Sin embargo, aun cuando la variedad y cantidad de factores humanos que inciden en la calidad del software es grande; y más aún, la incidencia de muchos no ha sido establecida del todo, existen investigaciones que individualmente y en contextos particulares confirman su incidencia. A continuación se detallan algunas de estas investigaciones, y se identifican, caracterizan y clasifican los factores humanos que se ha encontrado que tienen relación con la calidad del software.

3.1 Factores humanos en la calidad del software

3.1.1 Competencias

La norma ISO 10018 [34] define la competencia como la “*capacidad de aplicar los conocimientos y habilidades para lograr los resultados previstos*”, y para asegurarla la organización debe [7, p. 7]:

- 1) Determinar la competencia necesaria de conformidad con los requisitos del trabajo.
- 2) Proporcionar formación o tomar otras acciones para lograr dicha competencia,
- 3) Evaluar la eficacia de las acciones tomadas,
- 4) Asegurarse de que los empleados son conscientes de la importancia de sus actividades y su contribución a los objetivos de calidad, y
- 5) Mantener los registros apropiados de la educación, formación, habilidades y experiencia de cada individuo en la organización.

Según esto, las competencias se refieren a la capacidad de aplicar o usar un conjunto de conocimientos, destrezas y habilidades relacionadas, para realizar con éxito las funciones o tareas de un trabajo. Las competencias están determinadas por los *conocimientos* y las *habilidades*, dos de los activos más importantes de un individuo. El conocimiento es la “*información, entendimiento o habilidad que se obtiene de la experiencia o la educación*” [47] y la habilidad es la “*la capacidad de hacer algo que viene de la formación, la experiencia o la práctica*” [48].

¿Inciden las competencias en la calidad del software? Un reporte de 1990 del Buró de Ciencias de la Computación y Tecnología (CSTB, por sus siglas en inglés) citado por McConnell [25, p. v] establece que las ganancias más grandes en la productividad y la calidad de software vienen de la

codificación, unificación y distribución del conocimiento existente de las prácticas efectivas para desarrollarlo. No es posible hacer algo, especialmente el desarrollo de software, sin el conocimiento y las habilidades necesarias.

“Las personas que son efectivas para desarrollar software de alta calidad han pasado años acumulando docenas de técnicas, trucos y encantos mágicos. Las técnicas no son reglas. Un buen artesano conoce la herramienta correcta para el trabajo y sabe cómo usarla correctamente. Los programadores también. Entre más se aprende acerca de la programación más se llenará la caja mental de herramientas con herramientas analíticas y el conocimiento de cuando y como usarlas correctamente” [13, p. 51].

Como en toda disciplina, la habilidad para hacer algo y el esfuerzo invertido en una tarea, está determinado por las competencias individuales y por tanto varía enormemente de persona a persona. McConnell [25, pp. 682, 820] citando varios estudios (Sackman, Erikson y Grant [49], Curtis [50], Mills [51], DeMarco y Lister [52], Curtis y otros [53], Card [54], Boehm y Papaccio [55], Valett y McGarry [56], y Boehm y otros [57]) afirma que en el desarrollo de software se pueden establecer diferencias en la calidad del software y la productividad entre el mejor y el peor desarrollador con un orden de 10:1 en promedio, considerando como medidas de la calidad y la productividad: el tamaño, la velocidad, y la tasa de errores.

¿Cómo obtener las competencias necesarias? La educación, el aprendizaje, la formación, la investigación, la observación y la experiencia son los procesos para el desarrollo de conocimientos y habilidades. El individuo, en primera instancia, debe garantizar el nivel de educación mínimo requerido para el puesto; y la empresa, en segunda instancia, debe proporcionar formación o tomar otras acciones para mejorar o actualizar la competencia si fuese necesario.

La adquisición de competencias y habilidades es importante. Como bien dice Benjamín Franklin citado por Hunt y Thomas [58, p. 12]: *“Una inversión en el conocimiento siempre retorna los mejores beneficios”*. Esos beneficios pueden ser, entre otros, mayor satisfacción y motivación de los empleados, el incremento de la productividad, la mejora en la calidad del producto en el cual ese conocimiento es aplicado, el incremento de las utilidades, y la mejora en la competitividad de las empresas.

Es importante entonces que, los individuos inviertan en adquirir los conocimientos y habilidades para lograr las competencias que se requieren para el desarrollo de software, y que las empresas inviertan en mejorarlo o actualizarlo.

¿Qué conocimientos se requieren para desarrollar software? Muchas organizaciones han dado respuesta a esta pregunta, entre ellas:

- 1) La Sociedad de Computación del Instituto de Ingeniería Eléctrica y Electrónica (IEEE-CS, por sus siglas en inglés) ha desarrollado el *“Cuerpo de Conocimientos de la Ingeniería de Software”* (SWEBOK, por sus siglas en inglés) y la certificación *“Profesional Certificado en el Desarrollo de Software”* (CSDP, por sus siglas en inglés). El objetivo del SWEBOK es promover una visión consistente de la Ingeniería de Software en todo el mundo, especificar su alcance en relación a otras disciplinas (como las Ciencias de la Computación, la Gestión de Proyectos, la Ingeniería de Computadores y las Matemáticas), caracterizar las competencias de la disciplina, y proveer las bases para el desarrollo de planes de estudio y la certificación [23, p. xxxi].
- 2) El Instituto de Ingeniería de Software (SEI, por sus siglas en inglés) de la Universidad Carnegie Mellon (CMU, por sus siglas en inglés) ha elaborado otro *“Cuerpo de Conocimientos de la Ingeniería de Software”* [59]. En este documento se describen las competencias necesarias en el desempeño profesional de la Ingeniería de Software con el objetivo de asistir a las organizaciones en su definición y mejora.
- 3) La Asociación de Maquinaria Computacional (ACM, por sus siglas en inglés) en conjunto con la IEEE-CS ha descrito las disciplinas y especializaciones en el área de la computación [60] y

han definido un plan de estudio para cada una ellas [61]. Las disciplinas identificadas son: *Ciencias de la Computación, Ingeniería de la Computación, Sistemas de Información, Tecnologías de la Información* y la *Ingeniería de Software*.

TABLA NO. 12 ÁREAS DE CONOCIMIENTO DE LA INGENIERÍA DE SOFTWARE

SWEBOK	SEI
Requisitos de software	Requisitos de software
Diseño de software	Diseño de software
Construcción de software	Codificación de software
Pruebas de software	Pruebas de software
Mantenimiento de software	Operación y mantenimiento de software
Gestión de la configuración	Gestión de la configuración
Gestión de la ingeniería de software	Gestión de proyectos
Proceso de la ingeniería de software	Gestión de riesgos
	Gestión del conocimiento
	Gestión de procesos
Calidad del software	Gestión de la calidad del software
Economía del software	
Fundamentos de computación	Algoritmos y estructuras de datos
	Arquitectura de computadores
	Sistemas Operativos
	Lenguajes de programación
Modelos y métodos de ingeniería de software	
Fundamentos de ingeniería	
Matemáticas	Matemáticas
Práctica profesional	

Nota. Elaboración propia a partir de información en SWEBOK V3.0 Guide to the Software Engineering Book of Knowledge [23, p. xxxii] y A Software Engineering Body of Knowledge Version 1.0 [59, pp. 9-34].

¿Qué habilidades se obtienen estudiando las disciplinas en el área de la computación? A este respecto, la ACM dice que se logran dos tipos de habilidades: habilidades técnicas y habilidades profesionales [62].

TABLA NO. 13 HABILIDADES EN EL ÁREA DE LA COMPUTACIÓN

Habilidades técnicas	Habilidades generales
Resolución de problemas	Comunicación escrita
Niveles de abstracción	Dar presentaciones efectivas
Construcción de bases de datos	Dar demostraciones de productos
Uso de bases de datos	Negociación
Uso de herramientas de software	Buscar trabajo
Programación	Ser miembro de equipos
Uso de librerías de software	Necesidad de estar actualizado
Usos de las computadoras	Fluidez en la computación
	Aptitudes organizacionales

Nota. Elaboración propia a partir de información en Skills you'll learn if you study computing [62].

Viendo hacia atrás dos o tres décadas, los desarrolladores de software no tenían mucha interacción social, sin embargo, esta situación ha cambiado hoy en día. Ahora se espera que los desarrolladores de software tenga no solo conocimiento técnico sino también se espera tengan la habilidad para aprender, para trabajar en equipo y habilidades para expresarse de forma oral o escrita.

Un estudio patrocinado por Microsoft [63] analizó alrededor de 14.6 millones de anuncios de trabajo para identificar las habilidades requeridas en los mejores puestos de trabajo del futuro (hasta el 2020), entre ellos, el de “Especialista y Programador de Computadoras”. Las primeras diez habilidades identificadas en orden de importancia fueron:

- 1) Comunicación oral y escrita,
- 2) Orientación al detalle,
- 3) Microsoft Office,
- 4) Orientación al servicio al cliente,
- 5) Aptitudes organizacionales,
- 6) Resolución de problemas,
- 7) Planeamiento operativo,
- 8) Bilingüe o multilingüe,
- 9) Iniciativa y automotivación, y
- 10) Poder trabajar sin supervisión.

Contar con el conocimiento y las habilidades en un momento dado no es suficiente. Debe recordarse que tanto el conocimiento como las habilidades son activos que expiran, su valor disminuye con el tiempo.

El conocimiento de un profesional se vuelve obsoleto a medida que nuevas técnicas, lenguajes y entornos tecnológicos son desarrollados y en esa medida el valor de él para la organización también disminuye, a menos que dedique esfuerzo para mantenerlo al día y desarrollar más habilidades: aprendiendo un nuevo lenguaje, leyendo libros técnicos y no técnicos; tomando cursos en universidades, institutos, escuelas, centros de capacitación o eventos; participando en grupos de discusión; experimentando con diferentes entornos; y suscribiéndose a revistas o grupos de noticias [64, pp. 13-15].

En relación a la lectura de libros DeMarco y Lister [13, p. 11] revelan una realidad desalentadora: las estadísticas de hábitos de lectura muestran que el desarrollador de software promedio no posee un solo libro en el tema de su trabajo y nunca ha leído uno.

3.1.2 Experiencia

La experiencia se refiere a la *“habilidad o conocimiento que se obtiene al hacer algo”* o a la *“cantidad de tiempo que se ha pasado haciendo algo”* [65]. La experiencia viene a ser entonces una forma de conocimiento o habilidad derivados de la observación, participación y vivencia de un evento. Cada trabajador debe tener una experiencia mínima en el puesto o puestos similares, este tiempo incluye un período de práctica en la organización.

¿Influye la experiencia en la calidad del software? Estudios han demostrado que no existe correlación entre los años de experiencia y la calidad del software. Uno de esos estudios es el de Sackman, Erickson y Grant [49]. Se estudiaron programadores profesionales con un promedio de 7 años de experiencia, y aunque se encontraron diferencias individuales en los tiempos de codificación, depuración, tamaño de los programas y velocidad de ejecución, no se encontró ninguna relación entre los años de experiencia y la calidad del código o la productividad.

En otros campos, lo que se aprende para el trabajo hoy es casi seguro ayudará mañana. En el campo del software si el desarrollador no puede desligarse de los hábitos de programación que funcionaban en el pasado, la experiencia que se tenga será peor que no se tenga ninguna [25, p. 832].

3.1.3 Profesionalismo

El profesionalismo es *“la habilidad, el buen juicio o el comportamiento cortés que se espera de una persona que está formada o capacitada para hacer bien su trabajo”* [66].

¿Incide el profesionalismo en la calidad del software? Miller [67] en su artículo *“La calidad del software requiere profesionalismo y fortaleza”* inicia su respuesta preguntándose:

“¿Qué significa profesionalismo para los Ingenieros de Software? ¿Significa que están comprometidos a aprender y seguir buenas prácticas y dedicar sus energías a servir las necesidades del cliente con pasión y entrega o significa que quieren escribir código como mejor les parezca, sin proporcionar métricas o realizar revisiones o pruebas, y que no se les debe pedir que sigan procesos o que se preocupen por plazos?” [67].

Profesionalismo viene a ser una característica de la persona que desempeña un trabajo con pericia, aplicación, seriedad, honradez y eficacia. Y seguido afirma:

“La calidad consiste en hacer lo correcto, y hacerlo de la manera correcta. Por lo tanto, la calidad del software consiste en saber cuál es la tarea correcta, los plazos, las necesidades del cliente, y luego asegurarse de que el software es desarrollado y entregado a través de métodos que proporcionan la mejor oportunidad para lograr las metas. Cualquier cosa que interfiera con esos métodos es un impedimento para el aseguramiento de la calidad del software. Los objetivos no se les logran por accidente o por equipos que no funcionan juntos de manera efectiva y predecible” [67].

Puede afirmarse entonces que, el profesionalismo es importante para lograr la calidad del software. Pero también, como anticipa el mismo Miller, simplemente confiar en la profesionalidad de los Ingenieros de Software no es suficiente, se debe combinar con un compromiso de gestión que utilice procesos definidos para identificar, asignar, programar, completar y verificar el trabajo. Procesos como los definidos por la ISO 9001 o el Modelo de Madurez y de Capacidad Integrado (CMMI por sus siglas en inglés).

¿Cómo se logra el profesionalismo en los Ingenieros de Software? Independiente de la disciplina, los profesionales siguen una ruta de desarrollo profesional según los elementos que estén disponibles para dicha profesión. Para Ford y Gibbs [68] los elementos de las profesiones maduras son:

- 1) Educación profesional inicial,
- 2) Acreditación,

- 3) Desarrollo de habilidades,
- 4) Certificación,
- 5) Concesión de permisos,
- 6) Desarrollo profesional,
- 7) Sociedades profesionales, y
- 8) Código de ética.

La profesión de la Ingeniería de Software tanto a nivel nacional como internacional no cuenta con todos los elementos listados. A nivel nacional se tiene educación profesional inicial y oportunidades de desarrollo de habilidades en universidades, institutos tecnológicos y algunas empresas; no se cuenta con instancias de acreditación y certificación ni mucho menos de concesión de permisos; se tienen asociaciones profesionales de carácter general como la Asociación Salvadoreña de Ingenieros y Arquitectos (ASIA) y la Federación Salvadoreña de Ingenieros, Arquitectos y Ramas Afines (FESIARA), y de carácter específico como la Asociación Salvadoreña de Profesionales en la Computación (ASPROC). El único código de ética que existe es el publicado por ASIA: el "*Código de Ética Profesional de la Asociación Salvadoreña de Ingenieros y Arquitectos*" [69].

A nivel internacional, a través del esfuerzo conjunto de la IEEE-CS y la ACM se cuenta con la guía SWEBOK, los planes de estudio para las disciplinas en el área de la computación y el Código de Ética y de la Práctica Profesional de la Ingeniería de Software [70].

¿Cuál debe ser el comportamiento de los Ingenieros de Software profesionales? Miller [67] lista los siguientes aspectos:

- 1) Comportarse como profesional diciendo la verdad, conociendo su área, siendo un líder, siendo un solucionador de problemas, defendiendo el uso de buenas prácticas, tratando a los demás con respeto, etc.
- 2) Comprender que técnicas beneficiarán a la organización, saber cuánto esfuerzo puede invertirse en las iniciativas de calidad y saber cuándo permitir y fomentar la libertad para actuar.
- 3) Asegurar la cooperación en las mejoras de calidad, promover la moral y proporcionar apoyo.
- 4) Ayudar a la gerencia a reconocer los símbolos y comportamientos que caracterizan a los buenos desarrolladores, y a identificar incentivos, reconocimientos y castigos que serán efectivos para mejorar el proceso.
- 5) Saber cómo programar, esto es, conocer de lenguajes, bases de datos, sistemas operativos, software de control de versiones, revisores de sintaxis, herramientas para pruebas automatizadas, etc.
- 6) Identificar y priorizar las necesidades de capacitación y tomar la iniciativa en establecer un ambiente de aprendizaje.
- 7) Buscar buenas prácticas de calidad, establecer el camino para reconocerlas y elogiar los logros.
- 8) Entender cuando y como hacer cumplir los procesos, y reconocer que se requerirá profesionalismo, fortaleza y uso de la psicología para lograr las mejoras.
- 9) Guiar las revisiones, inspecciones, y los equipos asegurándose de que se llevan a cabo con eficacia.
- 10) No hablar mal de los desarrolladores o los procesos existentes, en su lugar, desarrollar líderes y utilizarlos como mentores.
- 11) Desarrollar una hoja de ruta de la calidad en la que se identifiquen las necesidades del negocio y sus beneficios.
- 12) No tomar modelos de procesos como un credo, si la organización realmente necesita CMMI o ISO 9001 abrir el camino, pero si no, ser honesto en las recomendaciones e implementar pequeñas mejoras graduales.
- 13) Asistir a conferencias profesionales, hacer presentaciones y publicar, y animar a otros también a hacerlo.

14) Convertirse en experto en el campo de desarrollo de software, la gestión y la calidad.

En el profesionalismo se incluye el compromiso, que se refiere a “*la actitud de alguien que trabaja muy duro para hacer o soportar algo*” [71].

¿Es el compromiso un factor determinante de la calidad del software? Cuando los individuos están completamente comprometidos en las actividades de la organización experimentan mayor satisfacción, y la organización en consecuencia conduce sus actividades más eficientemente.

En relación al compromiso de los niveles directivos, Siakas y Georgiadou [72] desarrollaron un estudio empírico para explorar la importancia del control total de la calidad y el papel del compromiso en la gestión de la calidad del software. En dicho estudio comentan que el compromiso en todos los niveles de la organización es primordial para lograr la transformación hacia el éxito en las organizaciones. El compromiso de la dirección de la empresa y el liderazgo son determinantes para motivar a los empleados en su esfuerzo continuo por la mejora de procesos y el logro de los objetivos de calidad.

Por otro lado, el compromiso de los empleados también es primordial para la mejora continua de los procesos, ya que el nivel de compromiso es la fuerza que permite perdurar en las dificultades del esfuerzo en la mejora de procesos, un esfuerzo que pueda ser considerado como una inversión y que provee resultados visibles años después [73].

3.1.4 Empoderamiento

Según la Asociación Americana de Calidad (ASQ, por sus siglas en inglés) el empoderamiento se refiere a dar los empleados los medios para tomar decisiones importantes y está basado en los conceptos de ampliación y enriquecimiento del trabajo. Ampliar el trabajo se refiere a añadir responsabilidades del mismo nivel jerárquico; y enriquecer el trabajo se refiere a añadir responsabilidades del nivel superior [74].

También la ASQ establece que el empoderamiento requiere:

- 1) El entrenamiento en las habilidades nuevas que el trabajo necesita,
- 2) El acceso a la información necesaria para tomar las decisiones, y
- 3) La iniciativa y confianza del empleado para tomar la nueva responsabilidad.

El empoderamiento es una práctica de gestión de compartir información, recompensas, y poder con los empleados para que puedan tomar la iniciativa y tomar decisiones para resolver problemas y mejorar el servicio y el rendimiento.

¿Incide el empoderamiento en la calidad del software? Si el proceso de empoderamiento se hace bien aumentará la productividad y la calidad. Productos de calidad se crean por desarrolladores competentes que no sólo siguen los procesos definidos, sino también están empoderados o facultados para mejorar esos procesos [74].

Además, programas de mejoras de procesos como el CMMI y el Modelo de Madurez y Capacidad para Personas (PCMM, por sus siglas en inglés) promueven el empoderamiento profesional como un aspecto crítico para asegurar la implementación apropiada de dichos modelos y lograr con ellos mejores niveles de calidad [75].

3.1.5 Conciencia de calidad

La conciencia se refiere al “*conocimiento de que algo existe (una situación, condición, o problema)*” [76]. La conciencia de calidad se refiere a comprender la pertinencia e importancia de las actividades que cada individuo realiza y de cómo estas contribuyen al logro de los objetivos de la calidad.

También, se debe entender que las actividades que se realizan no son actividades aisladas sino que son parte de un proceso y que están interconectadas, que si una de ellas falla afectará todo el proceso y no se conseguirá el resultado esperado.

¿Incide la conciencia de calidad en la calidad del software? La conciencia de calidad es importante para lograr un producto de calidad y el logro de esa conciencia de calidad debe ser asegurada por la organización [7, p. 7].

McConnell [25, p. 466] menciona que el establecer objetivos de calidad explícitos es una herramienta poderosa para mejorar la calidad del software, y que esa conciencia debe incluir objetivos tanto para los atributos externos de la calidad del software (corrección, usabilidad, eficiencia, confiabilidad, adaptabilidad, exactitud y robustez) como internos (mantenibilidad, flexibilidad, portabilidad, reusabilidad, legibilidad, facilidad de ser probado y comprensibilidad).

3.1.6 Creatividad

La creatividad se define como “*la habilidad para hacer nuevas cosas o pensar nuevas ideas*” [77]. La capacidad de generar nuevas ideas o conceptos, o de nuevas asociaciones entre ideas y conceptos conocidos, que habitualmente producen soluciones originales. Es sinónimo de pensamiento original, imaginación constructiva, pensamiento divergente o pensamiento creativo.

Aplicando la definición al desarrollo de software, puede decirse que el trabajo creativo es aquel trabajo o proceso que es percibido como útil para crear productos novedosos o para encontrar ideas y métodos nuevos y eficientes para resolver los problemas en el desarrollo de software [78].

¿Existe correlación entre la creatividad y la calidad del software? Según Glass, citado por Xin Tong [78], existe relación pues la actividad de desarrollo de software es principalmente una actividad de resolución de problemas complejos que requiere de la creatividad.

Por otro lado, según Cusomano, siempre citado por Xin Tong [78], demasiado énfasis en la creatividad y la independencia puede acarrear problemas en el control de los costos de desarrollo, calidad y mantenimiento a largo plazo; y demasiada estructura y control puede ahogar la creatividad, la innovación y la habilidad para cambiar.

¿Qué aspectos limitan la creatividad? Adams, uno de los principales maestros de diseño del mundo, citado por Godfrey [79], reconoce los siguientes tipos de obstáculos a la creatividad:

- a) *Obstáculos de percepción*: dificultad para identificar el problema, delimitar el problema demasiado, no ver los diferentes puntos de vista del problema, ver lo que uno espera ver (estereotipos), etc.
- b) *Obstáculos culturales o del entorno*: tabúes; idea de que la fantasía es una pérdida de tiempo; idea de que el humor, los sentimientos, la intuición, los juicios cualitativos son malos; falta de cooperación; falta de confianza; liderazgo autocrático; distracciones; falta de apoyo; etc.
- c) *Obstáculos emocionales*: miedo a cometer errores, fallar o arriesgarse; no poder tolerar la ambigüedad; preferencia por juzgar en lugar de proponer ideas; imposibilidad de relajarse; falta de interés; incapacidad de separar la realidad de la fantasía; etc.

3.1.7 Personalidad

La Asociación Americana de Psicología (APA, por sus siglas en inglés) define la personalidad como: “*las diferencias individuales en patrones característicos en la forma de pensar, sentir o comportarse*” [80]. Es el conjunto de rasgos con las que una persona nace y que pueden ser afectados por su desarrollo social y cultural.

Desde el inicio de los tiempos, los seres humanos han tratado de describirse y categorizarse de muchas formas. Las civilizaciones antiguas lo hicieron a través de los 4 temperamentos: *sanguíneo, colérico, melancólico y tranquilo*. Por su parte Carl Jung definió, para caracterizar la personalidad, los tipos de actitud (*extrovertido e introvertido*), las funciones perceptivas (*sensación o intuición*), y las funciones juzgadoras (*pensamiento y sentimiento*).

La teoría de Jung fue estudiada en 1920 por Myers y Briggs [81] quienes crearon el Indicador de Tipo de Myers-Briggs (MBTI, por sus siglas en inglés) que describía 4 rasgos de personalidad:

- 1) Introversión (I) o Extraversión (E).
- 2) Intuición (N) o Sensorial (S).
- 3) Pensamiento (T) o Emocional (F).
- 4) Juzgador (J) o Perceptivo (P).

También describieron, a partir de las posibles combinaciones de los 4 rasgos de personalidad, los 16 tipos de personalidad siguientes:

TABLA NO. 14 TIPOS DE PERSONALIDAD

Clase	Tipo	Abreviatura	Roles
Analistas	Arquitecto	INTJ	Pensadores, imaginativos y estratégicos, con un plan para todo.
	Lógico	INTP	Inventores, innovadores, con una sed insaciable por el conocimiento.
	Comandante	ENTJ	Líderes audaces, imaginativos y de voluntad fuerte, siempre en busca de un camino, o creando uno.
	Innovador	ENTP	Pensadores, inteligentes y curiosos que no pueden resistir un reto intelectual.
Diplomáticos	Abogado	INFJ	Callados y místicos, que sin embargo son inspiradores e idealistas incansables.
	Mediador	INFP	Personas poéticas, amables y altruistas, siempre en busca de ayudar.
	Protagonista	ENFJ	Líderes carismáticos e inspiradores, capaces de cautivar a quienes los escuchan.
	Activista	ENFP	Espíritus libres entusiastas, creativos y sociales, que siempre pueden encontrar una razón para sonreír.
Centinelas	Logista	ISTJ	Individuos prácticos y enfocados en hechos, de cuya confiabilidad no puede dudarse.
	Defensor	ISFJ	Protectores, muy dedicados y cálidos, siempre listos para defender a los suyos.
	Ejecutivo	ESTJ	Administradores excelentes, inigualables para administrar cosas y personas.
	Cónsul	ESFJ	Personas extraordinariamente consideradas, sociables y populares, siempre en busca de ayudar.
Exploradores	Virtuoso	ISTP	Experimentadores, audaces y prácticos, maestros en el uso de todo tipo de herramientas.
	Aventurero	ISFP	Artistas flexibles y encantadores, siempre listos para explorar y experimentar lo nuevo.
	Emprendedor	ESTP	Personas inteligentes, enérgicas y muy positivas, que realmente disfrutan vivir al límite.
	Animador	ESFP	Animadores espontáneos, enérgicos y entusiastas. La vida nunca es aburrida a su alrededor.

Nota. Elaboración propia a partir de información en Tipos de personalidad [82].

Pruebas de personalidad han sido usadas por muchos años como herramientas para encontrar individuos que se ajustan mejor a los requisitos de un puesto en particular, para identificar individuos con fortalezas y debilidades en determinadas características.

¿Incide la personalidad en la calidad del software? Existen muchos estudios, citados por Capretz [83], alrededor de la relación entre los tipos de personalidad y los Ingenieros de Software: Bush y Schkade (1985) identificaban que los tipos de personalidad más comunes en los ingenieros de software eran el ISTJ (Logista), seguido del INTJ (Arquitecto) y el ENTP (Innovador); Buie (1998) identificaba el tipo ISTJ (Logista), seguido del INTP (Lógico) y el INTJ (Arquitecto); Smith (1989) identificaba el tipo ISTJ (Logista) y ESTJ (Ejecutivo); Lyons (1985) identificaba el tipo ISTJ (Logista), seguido de INTJ (Arquitecto) e INTP (Lógico).

Por su parte, Capretz [83] encontró que el tipo ISTJ (Logista) era el tipo más común y considerando que el rasgo introvertido era el más frecuente concluía “*la mayoría de los ingenieros de software (ISTJ) son orientados técnicamente y prefieren trabajar con hechos y la razón más que con personas*”.

Un resultado opuesto encontró Varona (2011), siempre citado por Capretz, quien identificó el tipo ESTJ (Ejecutivo) como el más común, y aseguraba que la adaptabilidad, la comunicación y el manejo del estrés eran de las características más importantes en los Ingenieros de Software.

Raza, Capretz y Mustafa [84] estudiaron los diferentes perfiles de personalidad de Ingenieros de Software en Pakistán utilizando el MBTI e identificando sus preferencias en los atributos de calidad. Encontraron que el tipo de personalidad más común entre los Ingenieros de Software eran el ISTJ (Logista), ENTP (Innovador), ESTJ (Ejecutivo) y el ISFJ (Defensor). Y también que los atributos de calidad que estos ingenieros preferían son: la funcionalidad, la usabilidad, la eficiencia y la portabilidad.

Estos estudios describen los tipos de personalidad más comunes entre los ingenieros de software, pero no su relación con la calidad del software. Un estudio en esta línea es el de Acuña y otros [85] quienes encontraron correlación positiva entre el rasgo de extroversión y la calidad del software. Pero dicho estudio no es concluyente considerando que la extroversión, según las investigaciones indicadas, no es el rasgo más común en los Ingenieros de Software.

3.1.8 Carácter

El carácter se refiere al conjunto de rasgos que definen la integridad de la persona, el conjunto de valores que tiene y que son aprendidos de otros (padres, maestros, amigos, etc.) Por ejemplo: responsable, ordenado, honesto, independiente, amigable, etc. [86]

¿Incide el carácter en la calidad del software? Aunque no se han identificado estudios recientes que establezcan la relación entre el carácter y la calidad del software, según McConnell [25, p. 819] publicaciones como “*La programación considerada como una actividad humana*” (1965) de Edsger Dijkstra, “*La psicología de la programación de computadores*” (1971) de Gerald M. Weinberg, y “*Experimentos exploratorios en el comportamiento del programador*” (1975) de Ben Shneiderman legitiman el carácter del programador como un campo fructífero de investigación.

McConnell [25, pp. 820-830] llega a afirmar que el carácter del programador no está fuera de tema pues la interioridad del proceso de desarrollo hace del carácter un factor más decisivo que la inteligencia del programador, y que algunos rasgos de carácter del buen programador son:

- 1) **Humildad:** las personas que son los mejores programadores son las personas que se dan cuenta cuan poco conocen, y los peores son los que se rehúsan a aceptar que su intelecto nunca va a estar a la medida de la tarea. De hecho muchas buenas prácticas de programación tienen por objetivo reducir la carga cerebral: la descomposición en partes más pequeñas de comprender; las revisiones, inspecciones y pruebas para compensar el error humano; mantener las rutinas o funciones pequeñas; escribir los programas en términos del dominio del problema ocultando los detalles de implementación de bajo nivel; y el uso de convenciones.
- 2) **Curiosidad:** la información técnica y la del negocio cambia continuamente, si no se es lo suficientemente curioso para identificar mejores formas de hacer el trabajo pronto se terminará desactualizado. El estar conscientes de todo el proceso de desarrollo y lo que involucra, la experimentación, el estudiar técnicas para la resolución de problemas, el analizar y planear antes de actuar, el estudiar proyectos exitosos, el leer libros y revistas vorazmente, el afiliarse con otros profesionales y el tener un compromiso con el desarrollo profesional, son mecanismos que ayudan a ejercitar la curiosidad.
- 3) **Honestidad:** rehusarse a parecer un experto cuando realmente no se es, admitir los errores inmediatamente, tratar de entender un error en lugar de suprimirlo, entender el programa en

lugar de sencillamente compilarlo para ver si funciona, proveer reportes realistas del estado del proyecto, proveer estimados realistas y mantenerlos aun cuando se pida que sean ajustados, son ejemplos de actuaciones honestas.

- 4) **Comunicativo y cooperador:** los buenos programadores saben cómo trabajar y divertirse bien con otros. Escribir código es parte de ser un jugador de equipo, y se debe tener en mente la persona que deberá mantener el código al escribirlo.
- 5) **Disciplina:** para desarrollar un buen código se requiere de un proceso y de convenciones, y se debe ser disciplinado para seguirlas.
- 6) **Pereza:** la pereza puede manifestarse como el diferir tareas que no son placenteras, el hacer las tareas no placenteras rápidamente para que ya no estorben, y el escribir una herramienta para hacer la tarea no placentera y no tener que hacerla más. La primera forma de manifestación no es beneficiosa pero las dos últimas sí lo son. La otra cara de la moneda es el afán o el hacer el esfuerzo. El afán es esfuerzo innecesario, demuestra que se tiene la voluntad pero no que se está haciendo el trabajo, confunde movimiento con progreso. La tarea más importante en la programación es “pensar”, y cuando el programador “piensa” no parece ocupado.

También McConnell menciona rasgos de carácter [25, pp. 820-830] que no son tan relevantes:

- 1) **Persistencia, perseverancia o tenacidad:** la persistencia en el desarrollo de software no es siempre una buena cualidad, en su lugar debe promoverse el rediseñar, el probar una nueva alternativa, o el volver más tarde a ella.
- 2) **Emoción o entusiasmo:** el entusiasmo, el pasar noches y noches programando hasta encontrar la solución no es sustituto de la competencia.

3.1.9 Estado de ánimo

El estado de ánimo se define como la actitud o disposición emocional en un momento determinado.

¿Incide el estado de ánimo en la calidad del software? Graziotin, Wang y Abrahamsson [87] en su investigación sobre cómo los desarrolladores felices resuelven mejor los problemas, exponen estudios que afirman que tanto estados afectivos positivos como negativos promueven la creatividad, la que a su vez en un nivel apropiado ayuda en la resolución de problemas complejos. Mencionan que Sowden y Dawson encontraron que la cantidad de ideas creativas es impulsada por estados afectivos positivos; y por otro lado, estudios de George y Zhou, y Kaufmann y Vosburg han demostrado que estados afectivos negativos también afectan negativamente la creatividad. Lo que queda claro es que los estados afectivos mantienen una relación con la creatividad en la resolución de problemas y la creatividad a su vez incide en la calidad del software.

3.1.10 Estructura organizativa

El software es desarrollado por individuos o por equipos de trabajo organizados según una estructura organizativa. Entendiendo por estructura organizativa la “*disposición usualmente jerárquica de líneas de autoridad, comunicaciones, derechos y deberes*” que determina “*cómo se asignan los roles, el poder y las responsabilidades; como son controlados y coordinados; y cómo la información fluye entre los diferentes niveles de gestión*” [88].

¿Incide la estructura organizativa en la calidad del software? Hay poca evidencia empírica del efecto de la estructura organizativa en la calidad del software. En el 2007 Mathew, citado en el estudio de Lavallée y Robillard [36] sobre el impacto de los factores organizacionales en la calidad, decía que era sorprendente que se haya ignorado hasta esa fecha la investigación del impacto de la organización en la calidad.

Aún con tan poca evidencia empírica, Brooks en 1995, en su libro *el Mítico Hombre-Mes*, afirmaba que la calidad del software se ve afectada fuertemente por la estructura organizativa, debido a que, a

medida que las organizaciones se vuelven más complejas las comunicaciones se vuelven más difíciles y esto a su vez acarrea problemas que inciden en la calidad, como: atrasos en el cronograma del proyecto, la no adaptación de los empleados a sus funciones, errores o defectos en los programas, etc. Y añade que, la estructura organizativa es la medida radical a dichos problemas, porque busca reducir la cantidad de comunicación y coordinación a través de la división del trabajo y la especialización [14, p. 74].

Un estudio reciente que evalúa la relación entre la estructura organizativa y la calidad del software es el de Nagappan, Murphy y Basili [37], patrocinado por Microsoft. En este estudio se definen 8 métricas para evaluar la complejidad organizacional y como inciden en la calidad del software medida a partir de la tasa de errores. Dichas métricas son:

- 1) Número de desarrolladores que han modificado una aplicación y aún trabajan para la organización, bajo el supuesto que a más personas modificando el código, mayor será la tasa de errores, a causa de la mayor coordinación requerida, y menor la calidad.
- 2) Número de desarrolladores que han modificado una aplicación y ya no trabajan para la organización, bajo el supuesto que la rotación de personal afecta la retención de conocimiento y a su vez la calidad.
- 3) Frecuencia de modificaciones que ha sufrido una aplicación, bajo el supuesto que a más modificaciones, mayor la inestabilidad del código y menor la calidad.
- 4) Nivel organizacional del contribuyente mayor (que realiza más del 75% de las modificaciones), bajo el supuesto que entre más bajo el nivel jerárquico, más enfocado estará en sus actividades, comunicación y responsabilidad y mayor será la calidad.
- 5) Porcentaje de la organización que contribuye al desarrollo (personas del nivel del contribuyente mayor contra el total de empleados), bajo el supuesto que a menor porcentaje, más cohesivos son los contribuyentes, menor será la carga de comunicación y coordinación y mayor la calidad.
- 6) Porcentaje de modificaciones que ha realizado una sola organización a una aplicación, bajo el supuesto que a mayor número de modificaciones de una sola organización, más cohesivas son las contribuciones y mayor la calidad.
- 7) Porcentaje de personas del mismo nivel del contribuyente mayor que modifican una aplicación, bajo el supuesto que entre más difundida es la contribución a una aplicación, menor es la calidad.
- 8) Número de organizaciones diferentes que contribuyen a más del 10% de las modificaciones de una aplicación, bajo el supuesto que entre más organizaciones diferentes la modifiquen, menor será la calidad.

Aunque el estudio, como lo advierten los autores, no puede generalizarse a todos los entornos, los resultados indicaron una correlación fuerte entre la complejidad de la estructura organizativa y la calidad. Todas las correlaciones fueron fuertes y estadísticamente significativas con un 99% de confianza al predecir los errores.

3.1.11 Cultura organizacional

Taylor citado por April y Laporte [89] define la cultura como:

“Ese todo complejo que incluye los conocimientos, las creencias, el arte, la moral o valores, las leyes, las costumbres y cualesquiera otras capacidades y hábitos adquiridos por el hombre como miembro de la sociedad”.

La cultura organizacional comprende entonces los conocimientos, creencias, valores, costumbres y otros hábitos adquiridos como miembro de la organización.

¿Influyen la cultura de la organización en la calidad del software? Según April y Laporte [89] la cultura guía los comportamientos, actividades, prioridades y decisiones individuales y de la organización, y en ese sentido, afecta la calidad del software.

Un estudio de Lavallée y Robillard [36] sobre el impacto de los factores organizacionales en la calidad del software identificó los siguientes aspectos culturales que inciden negativamente en la calidad:

- 1) *Cultura de actuar según la necesidad y no según un plan*: se refiere a la práctica de implementar nuevos procesos cuando la necesidad o una crisis no esperada, como el fracaso de un proyecto de desarrollo, así lo demanda y no en base a un análisis anticipado de las necesidades y de las metas de la organización.
- 2) *Cultura de protección del presupuesto*: se refiere la idea de cuidar que el presupuesto no se incremente lo que lleva a preferir soluciones temporales en lugar de soluciones definitivas a los problemas, bajo la mentalidad de que “es más barato construir un contenedor que resolver el problema de una vez por todas” o “es más barato parchar que corregir”.
- 3) *Cultura de protección del alcance y plazos*: con el objetivo de proteger el alcance del proyecto y los plazos se recurre a la adición de recursos (con el consecuente aumento del presupuesto). Lo que no funciona, tal como lo expresa la Ley de Brooks que dice: “La adición de mano de obra a un proyecto retrasado lo retrasa más” [14, p. 25].
- 4) *Cultura de cabildeo*: se refiere a no seguir el proceso debido, sino recurrir a favores para negociar soluciones.
- 5) *Cultura de presión indebida*: prioridades impuestas por una autoridad superior como la alta dirección, el cliente o un compañero de trabajo respetado, en base a lo que ellos perciben como relevante y no necesariamente lo que es importante para el éxito del proyecto.
- 6) *Cultura de no compartir el conocimiento*: tener poco recurso humano que tiene conocimiento del proyecto y no promover la práctica de compartir y distribuir el conocimiento en todo el equipo lo pone en riesgo de sobrevivencia o de retrasos. Si algo les ocurriera a los únicos miembros del equipo que conocen el proyecto, este se vendría abajo. Una métrica interesante que mide este hecho es la métrica “número del tractor” atribuida a Jim Coplien y que dice: “¿Cuánto personal del proyecto tendrían que ser golpeado por un tractor (o irse) antes de que el proyecto se vea incapacitado?”.

Acuña y otros [85] encontraron que los siguientes factores del clima organizacional promueven la calidad del software:

- 1) *Confianza participativa*: cuanta confianza sienten los miembros del equipo en expresar sus opiniones e ideas.
- 2) *Orientación a la tarea*: cuanto esfuerzo el equipo pone en lograr la excelencia en lo que hace.

Mientras que no encontraron relación entre los siguientes factores y la calidad del software:

- 1) *Soporte a la innovación*: soporte dado al equipo para proponer y ejecutar ideas innovadoras.
- 2) *Visión del equipo*: cuan claro el equipo define objetivos y metas.

Por su parte Wiegers [90] afirma que la cultura de la organización es un factor crítico de éxito en la mejora de procesos y en la calidad del software. También menciona que, para lograr un software de calidad se debe promover una cultura organizacional que garantice:

- 1) La participación del cliente en el proceso de desarrollo de software,
- 2) El compartir la visión del producto final con el cliente,
- 3) La idea de la calidad como primera prioridad y de la productividad como su consecuencia natural,
- 4) La mejora continua del proceso de desarrollo de software,
- 5) El poder compartir ideas y prácticas con los compañeros de trabajo,
- 6) La educación constante para mantener los conocimientos y habilidades al día,
- 7) La detección interna de errores y no por parte del cliente,

- 8) La existencia de procedimientos escritos de desarrollo de software,
- 9) Una atmosfera de confianza y respeto mutuo,
- 10) Un proceso de desarrollo iterativo en todas las etapas del desarrollo de software,
- 11) El no aceptar incitaciones a realizar un mal trabajo por parte de jefes, compañeros o clientes,
- 12) El sentimiento de que el trabajo de cada individuo es notado y que es importante, y
- 13) El hacer lo que tiene sentido y no dar lugar a dogmas.

Los valores que permean la organización en especial, y la profesión en general, están también relacionados con la calidad. Un estudio realizado por Jaktman en 1998 [36] demostró que los valores de la organización influyen en la calidad, evidenciando problemas como código duplicado y alta tasa de errores.

Las organizaciones, usualmente establecen una visión, misión, principios corporativos, códigos de ética o de conducta, o códigos de buen gobierno corporativo con el objetivo de guiar la actuación de las personas y donde se mencionan valores como la excelencia, la integridad, la transparencia, la honestidad, el respeto, la justicia, etc. Muchos de esos valores están asociados con la calidad, especialmente el de la excelencia.

También las profesiones cuentan con códigos de ética, la Ingeniería de Software no es la excepción. A nivel nacional se tiene el “Código de Ética” de ASIA que aunque no menciona la calidad explícitamente, en su inciso 2 establece que, se debe mantener una conducta profesional y velar por un ejercicio correcto de la profesión y en su inciso 3 establece que, se debe buscar la constante superación y actualización de los conocimientos profesionales [69]. Aspectos que, como se mencionó, inciden en la calidad del software.

A nivel internacional se tiene el “Código de Ética y de la Práctica Profesional de la Ingeniería de Software” desarrollado conjuntamente por la IEEE-CS y la ACM que, en el inciso 3.0.1 establece que se debe luchar por la alta calidad del software [70]. También se menciona la calidad cuando se abordan las cualidades personales o los aspectos de gestión dejando la idea de la importancia de la calidad en el actuar personal y profesional.

3.1.12 Estilo de liderazgo

El liderazgo es:

“La actividad o la habilidad de dirigir un grupo de personas o una organización. El liderazgo comprende:

- a) *Establecer una visión clara,*
- b) *Compartir esa visión con los demás para que la puedan seguir de buen grado,*
- c) *Proporcionar la información, el conocimiento y los métodos para alcanzar esa visión, y*
- d) *Coordinar y el equilibrar los intereses en conflicto de todos los miembros y las partes interesadas” [91].*

El estilo de liderazgo puede definirse entonces como la forma en que el líder dirige a un grupo de personas, proporciona indicaciones, las motiva e implementa planes. Estilos de liderazgo son [92]:

- 1) **Directivo:** el líder dice al equipo que hacer y cómo hacerlo.
- 2) **Consultivo:** el líder busca el consejo del equipo antes de tomar una decisión sobre lo que debe hacerse.
- 3) **Participativo:** el líder se ubica como miembro y discute la posible decisión con el equipo.
- 4) **Negociador:** el líder utiliza incentivos para motivar al equipo a hacer ciertas cosas.
- 5) **Delegativo:** el líder deja que el equipo tome sus propias decisiones sobre lo que debe hacerse.

¿Incide el estilo de liderazgo en la calidad del software? Puede afirmarse que así es. Deming, quien inició el movimiento de Gestión Total de la Calidad (TQM, por sus siglas en inglés) lo menciona

como uno de los 14 principios para transformar la efectividad de las organizaciones. El principio dice que debe instaurarse el liderazgo en la organización y que “*el trabajo de un líder no es decirle a la gente qué hacer, ni castigarlos, sino dirigir. Dirigir consiste en ayudar a la gente a hacer un mejor trabajo y a aprender por métodos objetivos*” [93].

También, la ASQ [94] reconoce que la forma en que una organización es dirigida es uno de los aspectos que tiene más impacto en su desempeño e identifica dos tipos de liderazgos:

- 1) *El liderazgo estratégico*: se refiere a definir la visión y misión de la organización; desarrollar estrategias, estructuras y procedimientos; y crear sistemas técnicos y sociales que atiendan las necesidades de los clientes y los empleados.
- 2) *El liderazgo operacional*: se refiere a asegurar que los procesos de la organización se lleven a cabo diariamente con eficacia; supervisar el rendimiento; tomar medidas sobre aspectos que limiten las operaciones; y asegurar que los empleados entienden lo que hay que hacer y están provistos de la autoridad, el conocimiento y las habilidades para hacerlo.

Por su parte, Prifling [38], quien exploró los estilos de liderazgo en proyectos de desarrollo, afirma que, para establecer proyectos exitosos es esencial contar con líderes fuertes y con capacidades para dirigir. Citando estudios de Durham y Summer menciona que, el liderazgo es primordial para la efectividad de los equipos y que la falta de liderazgo o un liderazgo inefectivo están entre los principales factores que dificultan el éxito de los proyectos. Los estilos de liderazgo que Prifling identifica como garantes del éxito de los proyectos son:

- 1) *Liderazgo colaborativo*: estilo basado en el involucramiento físico y mental del líder en el equipo de proyecto. El líder colaborativo hace hincapié en la accesibilidad y tiene la voluntad de ayudar a sus subordinados.
- 2) *Liderazgo directivo*: estilo basado en el poder y respeto. El líder directivo proporciona indicaciones y expectativas claras, comunica la información de los hitos, plazos y entregables del proyecto a los subordinados.

En la misma línea, DeMarco y Lister [13, p. 171] sostienen que el liderazgo debe ser un servicio, e identifican otro tipo de liderazgo que debería practicarse en los equipos de desarrollo para ser efectivos:

“En los mejores equipos, diferentes individuos proveen liderazgo ocasional, haciéndose cargo de las áreas en las que tienen fortalezas concretas. Nadie es el líder permanente, porque esa persona entonces dejaría de ser un compañero y la interacción del equipo comenzaría a descomponerse”.

¿Cuáles son las características de un buen líder? DeMarco y Lister [13, p. 100] mencionan que el mejor liderazgo es aquel ejercido por alguien que: le pone ganas, es apto, se prepara anticipadamente, maximiza el valor para todos, y lo hace con humor y con buena voluntad.

Otras características y habilidades del buen líder son [95]:

- 1) **Formación y experiencia.**
- 2) **Habilidades orientadas a las tareas:** planear, organizar, tomar decisiones, delegar, compartir el poder, resolver problemas, facilitar el proceso y ser motivador.
- 3) **Habilidades interpersonales:** manejar conflictos, ser persuasivo, tener influencia, ayudar o tutorar, ser comprensivo y proveer ayuda.
- 4) **Habilidades de comunicación:** saber escuchar, comunicar información, proveer retroalimentación y comunicar una visión.
- 5) **Habilidades de conexión:** establecer redes y límites, rendir cuentas y ser responsable.
- 6) **Rasgos personales:** tener confianza en sí mismo y estabilidad emocional, ser consistente, confiable y flexible.

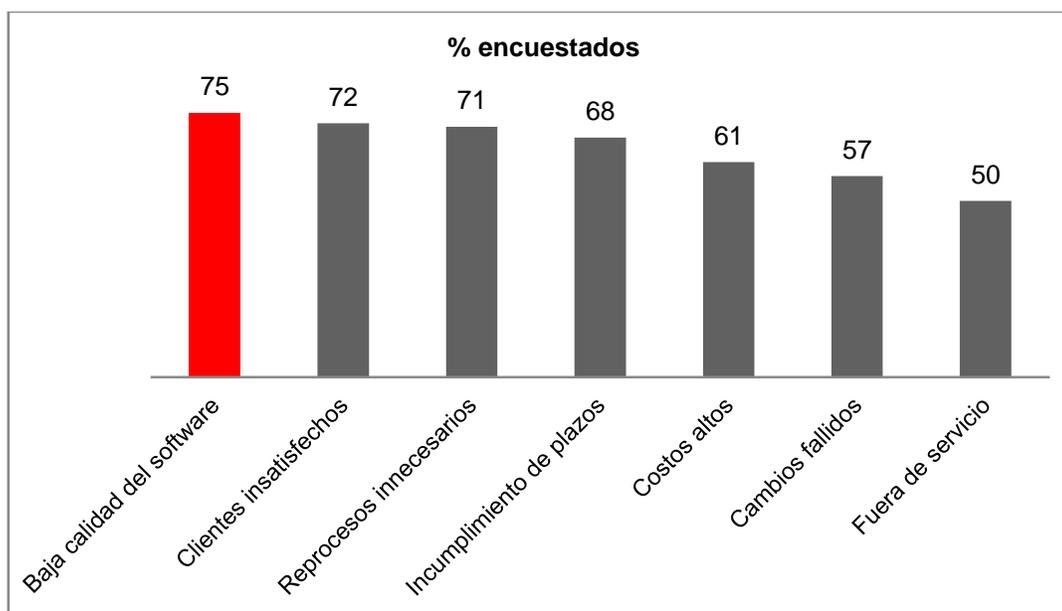
3.1.13 Gestión del cambio

La gestión del cambio se refiere a “*minimizar la resistencia al cambio organizacional a través del involucramiento de los participantes claves y los grupos de interés*” [96].

¿La gestión del cambio incide en la calidad del software? Según la ASQ un alto porcentaje de los cambios introducidos en las organizaciones no alcanzan su máximo potencial (no se implementan totalmente o no producen los beneficios previstos) debido a que se falla en la gestión del cambio: no reconocer la reacción normal de las personas de oponerse al cambio, no cultivar la disposición en lugar de la resistencia y no explicar el por qué deben querer cambiar [97]. La gestión del cambio es de suma importancia para el logro de los objetivos de la organización y por tanto para el logro de los objetivos de calidad, que a su vez influenciarán la calidad del software.

Un estudio realizado por Forrester Consulting [98] sobre los retos de la gestión del cambio del software expone que la gestión del cambio debe ser una práctica central para toda organización de tecnologías de la información y que aquellas que descuidan este aspecto sufren problemas en áreas críticas, entre ellas la baja calidad del software.

FIGURA NO. 8 PROBLEMAS CAUSADOS POR NO GESTIONAR EL CAMBIO



Nota. La figura muestra que el principal problema identificado como consecuencia de no gestionar apropiadamente el cambio es la baja calidad del software. Adaptado de *The Challenges Of Software Change Management In Today's Siloed IT Organizations* [98].

¿Qué tan fácil o qué tan difícil puede resultar la introducción de cambios en la organización?

La facilidad o dificultad está determinada por la cultura organizacional, uno de sus principales inhibidores. Los cambios no pueden ordenarse, tienen que ser aceptados y asumidos por todos los miembros de la organización y tienen que ser gestionados [89].

3.1.14 Gestión del conocimiento

La gestión del conocimiento se refiere a las:

“Estrategias y procesos diseñados para identificar, capturar, estructurar, valorar, usar eficientemente y compartir los activos intelectuales de una organización para mejorar su rendimiento y competitividad. Se basa en dos actividades fundamentales: 1) captura y

documentación del conocimiento explícito y tácito individual, y 2) su difusión dentro de la organización” [99].

Significa entonces identificar, usar y compartir de forma eficiente los activos intelectuales de una organización para mejorar su rendimiento y competitividad; y tiene por objetivo garantizar que el conocimiento se comparte dentro de la organización y se utiliza con eficacia.

¿Incide la gestión del conocimiento en la calidad del software? Basili y Caldiera [39] sostienen que, el reusar el conocimiento, los productos de software y la experiencia de hecho incrementan la calidad del software y proponen un enfoque que ellos llaman “*fábricas de experiencia*”: “organizaciones que soportan la reutilización de la experiencia y el aprendizaje colectivo desarrollando, modificando y proveyendo grupos de competencias a ser usados en los proyectos”.

Y añaden que una organización puede gestionar la calidad del software de dos formas:

- 1) Mejorando la efectividad del proceso de desarrollo de software reduciendo la cantidad de trabajo que se rehace y reusando artefactos de software en las etapas de un mismo proyecto o entre proyectos, y
- 2) Desarrollando e implementado planes para una mejora controlada, sostenida y continua basada en hechos e información.

También indican que las organizaciones deben capitalizar los éxitos y aprender de las fallas con una visión ampliada del proceso de desarrollo de software y que al establecerse como “*fábricas de experiencia*” se benefician a través de:

- 1) El establecimiento de un proceso de mejora del software sustentado y controlado por información cuantitativa,
- 2) La producción de un repositorio de información y modelos que están basados en la práctica y experiencia diaria,
- 3) El desarrollo de un soporte interno que reduce los gastos generales y provee beneficios sustanciales en el costo y la calidad,
- 4) La provisión de un mecanismo para identificar, evaluar e incorporar en el proceso nuevas tecnologías que han probado ser valiosas en contextos similares, y
- 5) La incorporación y soporte de la reutilización en el proceso de desarrollo de software.

TABLA NO. 15 VISIONES DEL DESARROLLO DE SOFTWARE

Visión tradicional	Visión ampliada
Entrega de productos y servicios específicos	Desarrollo de capacidades
Descomposición de un problema complejo en otros más simples	Unificación de diferentes soluciones en una más general
Diseño e implementación	Análisis y síntesis
Detalle	Abstracción a partir del detalle
Validación y verificación	Experimentación

Nota. Tomado de *Improve Software Quality by Reusing Knowledge and Experience* [39].

Reusar el conocimiento, productos y experiencia permite a las empresas mantener altos niveles de calidad porque los desarrolladores no necesitan estar adquiriendo constantemente nuevo conocimiento y pericia cuando trabajan en diferentes proyectos [100].

3.1.15 Trabajo en equipo

El trabajo en equipo se refiere a la habilidad de trabajar con otros. El software se desarrolla en equipos de variadas características, entre ellas: el tamaño, la distribución, las competencias y habilidades de los miembros, y la cohesión.

¿Qué características del equipo inciden en la calidad del software? La calidad del software es en gran medida dependiente del buen trabajo en equipo. Hoegl y Gemuenden estudiaron la influencia de seis factores de calidad del trabajo en equipo: comunicación, coordinación de habilidades, balance de la contribución de los miembros, soporte mutuo, esfuerzo y cohesión. Los seis factores de calidad demostraron estar fuertemente relacionados con la productividad [101].

Comunicación del equipo. La comunicación es un factor cuyo papel es central al desarrollo de software y se refiere a *“el acto o proceso de usar palabras, sonido, signos o comportamientos para expresar o intercambiar información o para expresar ideas, pensamientos o sentimientos a alguien más”* [102].

Como se mencionó anteriormente, gran parte de los problemas en los proyectos de desarrollo que inciden en la calidad del software se originan en problemas de comunicación. Ejemplos de estos problemas son: atrasos en el cronograma del proyecto, la no adaptación de los empleados a sus funciones, errores o defectos en los programas, etc. [14, p. 74]

También es conocido, que a medida que el número de personas en un proyecto aumenta así aumenta la complejidad de las comunicaciones y el riesgo de errores de comunicación, dado que, aumenta el número de caminos de comunicación.

McConnell [25, p. 650] explica que el número de caminos de comunicación aumenta de acuerdo a la siguiente expresión:

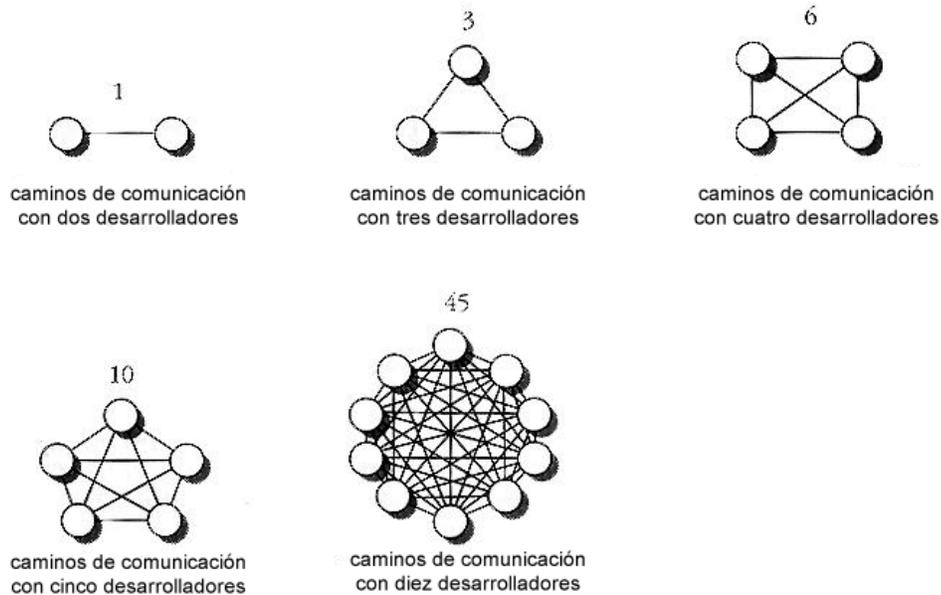
$$r = n * \frac{n - 1}{2}$$

Dónde: r = número de caminos de comunicación.

n = número de personas en el equipo.

Según esta fórmula un equipo de dos personas tiene 1 camino de comunicación, un equipo de 3 personas tiene 3, un equipo de 4 personas tiene 6 y así sucesivamente.

FIGURA NO. 9 LÍNEAS DE COMUNICACIÓN POR TAMAÑO DE EQUIPO



Nota. Tomado de Code Complete 2 [25, p. 650].

Un estudio realizado en los Laboratorios Bell y reportado por Boehm en 1981 explica porque no resulta extraño que la mayor parte de los problemas en el desarrollo de software se originen en problemas de comunicación: *los desarrolladores pasan la mayor parte del tiempo comunicándose.*

Lo que resulta interesante en este estudio, es que, esas actividades de comunicación se originen de actividades no técnicas [25, p. 681].

TABLA NO. 16 DISTRIBUCIÓN DEL TIEMPO DE LOS DESARROLLADORES

Actividad de comunicación	Tipo de actividad (%)				Total
	Personal	Administrativas	Desarrollo	Capacitación	
Conversando	7	21	5		33
Hablando por teléfono	1	2			3
Leyendo		2	14	2	18
Escribiendo		1	13		14
Afuera	1	8		6	15
Caminando	1	3	2		6
Otros	3	4	4		11
Total	13	41	38	8	100

Nota. Estudio de tiempos y movimientos de 70 desarrolladores en los Laboratorios Bell en 1964 por Bardain y reportado en 1981 por Boehm. Adaptado de Code Complete 2 [25, p. 681].

Coordinación y balance del equipo. La coordinación y el balance del equipo se refieren al desarrollo y acuerdo de una estructura de tareas y responsabilidades entre los miembros del equipo, siendo la gestión de las habilidades y experiencia de cada uno de ellos el punto clave [101]:

- 1) Conocer donde las habilidades están situadas dentro del equipo,
- 2) Reconocer la necesidad de otras habilidades, y

3) Llevar esas habilidades a buen uso.

Vinod [103, pp. 3, 38] analiza el impacto de las habilidades del equipo de desarrollo en la calidad del software. Citando estudios de Richard, Robert, Justin, Evans, Krishnan y Boehm afirma que el factor determinante de la calidad del software son las personas que conforman el equipo que lo desarrolló, tanto de sus habilidades técnicas como de sus habilidades de gestión.

Las habilidades técnicas comprenden [103, pp. 41-45]: el dominio de la aplicación o experiencia del desarrollador de software en el dominio del negocio, el conocimiento y experiencia en el hardware y el software en que residirá la aplicación, el conocimiento y experiencia en el lenguaje de programación en el que estará escrita la aplicación, y la habilidad para programar e implementar la solución de software.

Y las habilidades de gestión comprenden [103, pp. 45-47]: el liderazgo o habilidad de influenciar a otros para lograr los objetivos compartidos, la motivación del equipo que determinará la intensidad, dirección y persistencia del esfuerzo hacia la consecución de un objetivo, y la gestión del proceso de comunicación del equipo y su involucramiento.

Soprote mutuo y cooperación del equipo. La calidad del software depende también de la cooperación del equipo porque el proceso de desarrollo de software es un proceso creativo que involucra la interacción humana, y dado que la idea de trabajo en equipo está basada en la idea misma de cooperación de los miembros más que la competencia entre ellos [101].

Cohesión del equipo. La cohesión del equipo se refiere al grado en que los miembros del equipo contribuyen a la habilidad del grupo de funcionar como una unidad. Equipos cohesivos tienen vínculos emocionales y sociales que los vinculan unos a otros y les motiva por las metas comunes.

Según Sommerville [12, p. 607] un equipo con las habilidades necesarias no es suficiente para que sea exitoso, el equipo debe ser cohesivo, lo que significa que el equipo debe tener un espíritu de grupo y estar motivado por las metas comunes más que por las personales. Y añade que un equipo cohesivo tiene las siguientes características:

- 1) Son leales y protegen al grupo como entidad,
- 2) Comparten las metas y objetivos del grupo,
- 3) Pueden establecer sus propios estándares y los respetan,
- 4) Aprenden de los demás y se apoyan mutuamente,
- 5) Comparten el conocimiento, lo que permite la continuidad en caso de retiro de un miembro, y
- 6) Promueven la mejora continua sin importar quién creó originalmente el diseño o el programa.

El estudio de Nagappan, Murphy y Basili [37] mencionado anteriormente, también concluye que la calidad disminuye a menor cohesión del equipo de desarrollo.

Tamaño del equipo. Somerville [12, p. 607] asegura que el tamaño del equipo incide en la calidad y que los equipos de desarrollo no deberían contar con más de 10 miembros, ya que, grupos pequeños reducen los problemas de comunicación.

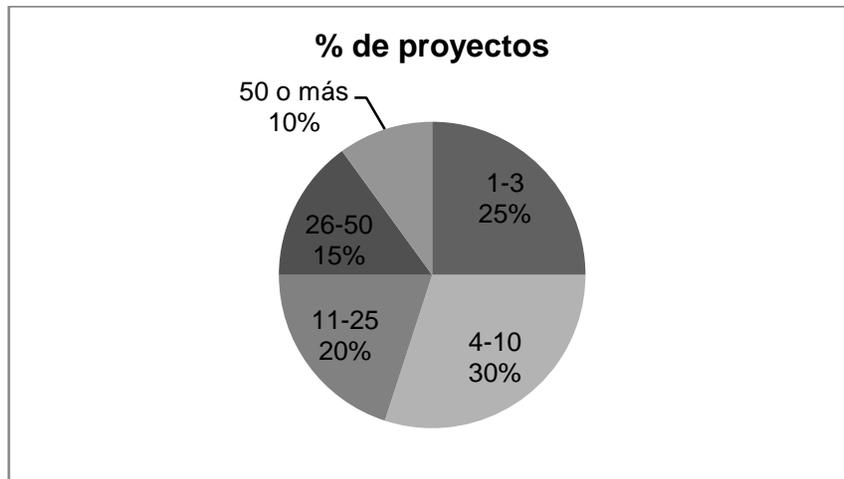
Un estudio, que establece una posible relación entre el tamaño del equipo y la calidad, es el de Mahajan [41]. El estudio buscaba establecer la relación entre el tamaño del equipo y la varianza de calidad del proyecto de desarrollo, esto es, la diferencia entre la calidad real valorada por las autoridades del proyecto y la calidad esperada. Los resultados indicaron que existe una pequeña correlación positiva entre ambas variables.

Otro estudio, aunque específico para el desarrollo de software de fuente abierta, es el de Meneely y Williams [40] quienes encontraron que más de 9 desarrolladores contribuyendo en el código fuente de una aplicación aumenta en 16 veces la probabilidad de incluir vulnerabilidades de seguridad. Este estudio puede considerarse no concluyente, pues en ambientes de desarrollo de software de fuente abierta, los equipos de desarrollo carecen de dirección formal.

También, Nagappan, Murphy y Basili [37] en su estudio sobre la influencia de la estructura organizativa en la calidad determinaron que la calidad disminuye a mayor número de personas modifican el código por el incremento en la complejidad de la coordinación requerida, y a mayor rotación de personal pues afecta la retención y transferencia del conocimiento de los requerimientos del negocio y el diseño de la aplicación.

En cuanto al número de miembros más común en los equipos de desarrollo estudios realizados por Beck, Perkins, Highsmith, Boehm y Turner analizados por McConnell [25, p. 651] mencionan que es de 4 a 10 miembros.

FIGURA NO. 10 TAMAÑO PROMEDIO DE LOS EQUIPOS DE DESARROLLO



Nota. Adaptado de Code Complete 2 [25, p. 651].

Distribución del equipo. La distribución del equipo se refiere a la localización de los miembros del equipo de desarrollo. Por ejemplo: en el mismo lugar o distribuidos en diferentes lugares (oficinas, departamentos, países, continentes, etc.)

Modelos de distribución del equipo son:

- 1) **Co-ubicados:** los miembros trabajan en el mismo lugar.
- 2) **Globalizados o distribuidos:** los miembros trabajan distribuidos geográficamente.
- 3) **Externalizados:** equipos con responsabilidades diferentes que trabajan fuera de la organización-

El desarrollo de software distribuido es una estrategia que cada vez está siendo más adoptada ya que ofrece ventajas en la reducción de costos y mejora de la productividad. Sin embargo, la comunicación entre los miembros del equipo de desarrollo se ve obstaculizada por la distancia física volviendo más compleja la dirección y coordinación de los proyectos [104].

Un estudio realizado conjuntamente por la Universidad de California, la Universidad de Zurich y Microsoft [42] evaluó la tasa de errores en el desarrollo de Windows Vista en equipos distribuidos y equipos locales. El estudio consideró:

- 1) Un proyecto de una sola organización, no un entorno externalizado o subcontratado.
- 2) Un proyecto de desarrollo a gran escala, compuesto de 4,000 archivos binarios y 3,000 desarrolladores.
- 3) Diferentes niveles de separación: edificio, campus, localidad, continente, etc.
- 4) Diferentes complejidades y características de mantenimiento.

El resultado del estudio determinó que la cantidad de errores era ligeramente mayor en los equipos distribuidos pero no lo suficientemente significativa como para concluir que la distribución del equipo incide en la calidad del software.

En dicho estudio se mencionan los siguientes aspectos como retos de los ambientes distribuidos:

- 1) *Dificultades de comunicación*: la comunicación se ve afectada debido a la falta de reuniones o su informalidad, miembros que no se conocen, disminución de la comunicación síncrona debido a las barreras de la zona horaria y el lenguaje, etc.
- 2) *Dificultades de coordinación*: la coordinación se vuelve más compleja debido a la falta de comunicación, menor conciencia del grupo, dificultad de los directores del proyecto de mantenerse al tanto de las tareas de cada miembro y como están relacionadas, etc.
- 3) *Diversidad de entornos tecnológicos*: la infraestructura tecnológica, las metodologías, métodos y herramientas en cada sitio y su uso de ellos.
- 4) *Propiedad del código*: un problema con los ambientes distribuidos es la propiedad distribuida, pues puede darse el caso que no esté claro quién es el responsable de realizar una tarea de prueba, modificación, etc.
- 5) *Diversidad de entornos de negocios*: usualmente hay relación entre la organización y entidades externas, como el gobierno o proveedores, que serán diferentes de una ubicación a otra.
- 6) *Escenarios de relación entre sitios*: si se trata de alianzas estratégicas entre las empresas, o de subcontratación o externalización de servicios, etc.
- 7) *Barreras culturales y organizacionales*: los compañeros de trabajo debe estar conscientes de las diferencias culturales, comportamientos de comunicación, etc.

Otro estudio que confirma la poca relación que existe entre la distribución del equipo y la calidad del software es el de Spinellis [105] quien examinó el efecto de la distancia geográfica de los desarrolladores en la productividad, el estilo de codificación y la densidad de defectos en el código base de FreeBSD.

¿Qué otros aspectos del trabajo en equipo inciden en la calidad del software? En entrevistas realizadas por Krasner citadas por Hemstra [16, p. 56] a desarrolladores y a directores de proyectos se obtuvo respuesta a esta pregunta.

Los desarrolladores mencionaron los siguientes aspectos:

- 1) Problemas que se llevan a discusión pero que no son tomados en serio,
- 2) Ser evaluados en base a criterios no relacionados con los objetivos del proyecto,
- 3) Que motivaciones individuales dominen los objetivos del equipo,
- 4) No poder opinar sobre los objetivos del proyecto y los métodos de trabajo,
- 5) Que los objetivos del proyecto no sean claros,
- 6) Imprecisión en relación a la autoridad y las responsabilidades,
- 7) Composiciones de equipo cambiantes,
- 8) Discusiones interminables acerca del alcance del proyecto, y
- 9) Problemas de comunicación dentro del equipo y entre el equipo y la organización.

Y los directores de proyecto mencionaron los siguientes aspectos:

- 1) Falta de profesionalismo en los miembros del equipo,
- 2) Insuficiente personal,
- 3) Estimaciones dadas por los miembros del equipo sin ninguna explicación de su fundamento,
- 4) Plazos dados por los miembros del equipo antes de que el trabajo haya sido especificado apropiadamente,
- 5) Manejo inapropiado de los cambios, y
- 6) Carencia de entrenamiento.

3.1.16 Motivación

La motivación es la “*la condición de estar entusiasmados por actuar o trabajar*” [106]. La motivación es interna al individuo, varía de acuerdo al objetivo y tienen una intensidad y duración, además de ser decisiva para el comportamiento humano [104].

Según diferentes teorías motivacionales, la motivación está estrechamente ligada a [104]:

- 1) La satisfacción de las necesidades (Teoría de Maslow).
- 2) Factores internos al individuo como el sentido de superación y externos como los incentivos, reconocimientos, beneficios y castigos (Teoría de Herzberg).
- 3) La búsqueda del significado de la vida (Logoterapia de Victor Frankl).
- 4) El empoderamiento (Teoría de William Byham).

¿Incide la motivación en la calidad en el desarrollo de software? El papel de la motivación en la calidad del software es indiscutible. Si las personas no están motivadas no estarán interesadas en la actividad que realizan, trabajarán con lentitud, y será más probable que comenten errores, y que no contribuyan a los objetivos de la organización [12, p. 603].

Estudios han demostrado que la motivación es uno de los factores más importantes para la productividad y la calidad. Además muchos de los otros factores humanos están directa o indirectamente relacionados con la motivación. El sentido de pertenencia, la seguridad en el trabajo, los premios e incentivos, el estilo de liderazgo y un balance adecuado de la vida laboral son considerados como promotores intrínsecos de la productividad y la calidad; mientras que la formación de equipos, la confianza y el respeto, la participación del empleado y el aprecio son considerados como promotores extrínsecos [107].

3.1.17 Satisfacción

Se entiende la satisfacción como “*el acto de proveer lo que se necesita o se desea*” [108]. En general se refiere al grado en que el individuo siente que sus necesidades (fisiológicas, de seguridad, sociales, de estima y de auto-realización) están satisfechas.

¿Incide el nivel de satisfacción en la calidad del software? Según un estudio de Graziotin, Wang, y Abrahamsson [109] afirma que empleados satisfechos producen software de mejor calidad. Los resultados del estudio ofrecen apoyo a la afirmación de que desarrolladores felices son mejores solucionadores de problemas en términos de sus capacidades analíticas.

Mencionan también algunas estrategias para lograr software de bajo costo pero de alta calidad: la asignación de los desarrolladores a oficinas privadas, la creación de un entorno de trabajo para apoyar la creatividad y la provisión de incentivos, todo aquello que haga a los desarrolladores más satisfechos y felices.

Samuelsson [110] observó también que desarrolladores satisfechos con sus condiciones de trabajo están más motivados y trabajan más eficientemente.

Según Maslow citado por Sommerville [12, p. 603] las necesidades se ordenan en una serie de niveles: en el nivel más bajo están las necesidades fisiológicas, luego las de seguridad, las sociales, las de estima y finalmente las de autorrealización.

FIGURA NO. 11 JERARQUÍA DE NECESIDADES HUMANAS



Nota. Tomado de Ingeniería de Software [12, p. 604].

Sommerville añade que las necesidades fisiológicas se refieren a las necesidades fundamentales de alimentación, sueño, etc.; las necesidades de seguridad se asocian al hecho de sentirse seguro en un entorno; las necesidades sociales con el hecho de sentirse parte de un grupo social; las necesidades de estima con el hecho de sentirse respetado; y las necesidades de autorrealización tienen que ver con el desarrollo personal.

Para cubrir las necesidades fisiológicas y de seguridad es necesario dar a los empleados salarios y beneficios justos. Salarios y beneficios que no cubren las necesidades básicas son causa de insatisfacción, lo que puede resultar en una baja productividad y baja calidad [107].

Para cubrir las necesidades sociales es preciso dar a los empleados tiempo para reunirse con sus compañeros y acceso a medios que faciliten las comunicaciones (como correo electrónico y redes sociales). La posibilidad de interactuar directamente con otros hace que las personas se conviertan en un grupo social y que acepten los objetivos y metas compartidas de la organización [12, p. 604].

Para cubrir las necesidades de estima es necesario demostrar a las personas que son valoradas y que su trabajo es notado e importante, factor que contribuye a la calidad del software [90]. El reconocimiento de los logros son también otra forma de cubrir las necesidades de estima [12, p. 604].

También es importante considerar lo que menciona DeMarco y Lister [13, p. 19] sobre cómo los individuos tienden a vincular la autoestima a la calidad del producto que producen. Les provoca poca satisfacción trabajar en cosas mediocres aunque pueda ser justo lo que se necesita para una situación dada. Así que, cualquier medida que ponga en peligro la calidad del producto causará insatisfacción, y un individuo insatisfecho no trabaja bien.

Finalmente, para cubrir las necesidades de autorrealización es necesario dar responsabilidades, asignarles tareas demandantes (pero no imposibles) y ofrecer un programa de formación donde puedan desarrollar sus habilidades; lo cual, además de ser un elemento motivador, redundará en beneficios en la calidad del software al incrementar las competencias del individuo.

3.1.18 Sentido de pertenencia

El sentido de pertenencia es una necesidad humana como la comida y el refugio. Puede definirse como el sentimiento de aceptación como miembro de una comunidad. El sentido de pertenencia aumenta la motivación, la salud y la felicidad [111].

¿Incide el sentido de pertenencia en la calidad del software? Según un estudio realizado por Sajid, Nasir, Shehzad y Mehtab [112] los empleados que se sienten parte de la organización y que sienten que están haciendo una contribución a su crecimiento y éxito les motivará, y estarán dispuestos a ir más allá de los límites para ayudar a que su organización tenga éxito.

Se debe promover el orgullo en las personas a través del empoderamiento, y la lealtad y *sentido de pertenencia*, promoviendo la responsabilidad [107].

3.1.19 Propiedad del código

La propiedad se refiere al “*acción y efecto de apropiar o apropiarse*” [113], esto es, hacer algo propio o adueñarse de algo.

La propiedad del código se refiere a si una persona es responsable de un componente de software o de su código (sea este un archivo, una clase, un módulo, o un subsistema) o si no hay nadie claramente responsable.

Modelos de propiedad del código son:

- 1) **Propiedad débil:** el sistema es asignado a un desarrollador, pero se permite la modificación por otros bajo la supervisión del dueño.
- 2) **Propiedad fuerte:** el sistema es propiedad de un solo desarrollador y solo él puede modificarlo.
- 3) **Propiedad colectiva:** cualquiera puede hacer modificaciones.

¿Incide la propiedad del código en su calidad? ¿Es mejor tener un solo responsable de un componente de software? Estudios indican que efectivamente así es. Uno de ellos, realizado por Bird, Nagappan, Murphy, Gall y Devanbu [114] en un estudio patrocinado por Microsoft encontraron que:

- 1) A mayor número de contribuyentes menores (que han hecho menos del 5% de los cambios), menor es la ocurrencia de errores previo o posterior a la puesta en producción.
- 2) A mayor propiedad por parte del principal contribuyente mayor (que ha hecho más del 5% de los cambios), menor es la ocurrencia de errores.
- 3) La propiedad del código tiene una correlación más fuerte con la ocurrencia de errores previo a la puesta en producción que posterior.

En dicho estudio, la propiedad del código se midió utilizando dos medidas:

- 1) Determinando la cantidad de desarrollo para un componente que proviene de un desarrollador. Si un desarrollador hace 80% de los cambios en un componente entonces tiene alta propiedad del código.
- 2) Determinando cuantos contribuyentes menores están trabajando en un componente. Si muchos desarrolladores hacen cambios a un componente entonces tiene baja propiedad del código.

3.1.20 Entorno ambiental

El entorno ambiental se refiere a los objetos y condiciones, tangibles y materiales que rodean el ambiente de trabajo.

¿Incide el entorno ambiental en la calidad del software? El ambiente físico hace una gran diferencia en la productividad y por tanto en la calidad. DeMarco y Lister realizaron un estudio en el que establecieron que programadores que hacen su trabajo mejor tienen oficinas más grandes, sin mucho ruido y privadas, y con menores interrupciones de personas o llamadas telefónicas [25, pp. 740-741].

TABLA NO. 17 CARACTERÍSTICAS DEL ENTORNO AMBIENTAL DE LOS MEJORES Y PEORES PROGRAMADORES

Característica del entorno ambiental	Mejores programadores	Peores programadores
Espacio físico dedicado	78 pies cuadrados	46 pies cuadrados
Quietud (ruido aceptable)	57% si	29% si
Espacio privado	62% si	19% si
Habilidad de silenciar el teléfono	52% si	10% si
Habilidad de redirigir llamadas	76% si	19% si
Interrupciones frecuentes	38% si	76% si
Ambiente que lo hace sentir apreciado	57% si	29% si

Nota. Adaptado de Code Complete 2 [25, pp. 740-741].

Los datos muestran una correlación fuerte entre la productividad y la calidad del entorno ambiental de trabajo.

3.2 Clasificación de los factores humanos

Los factores humanos de la calidad del software descritos se han organizado en 3 categorías:

- a) **Factores individuales**, son aquellos que se originan del individuo como los relacionados con las competencias y la experiencia, la personalidad y el carácter, por ejemplo.
- b) **Factores organizacionales**, son aquellos que se derivan del entorno organizacional en el que el individuo trabaja como los relacionados con la estructura organizativa, la cultura organizacional y el liderazgo, por ejemplo.
- c) **Factores de involucramiento en el trabajo**, son aquellos que se refieren a la participación del individuo en el trabajo o de su contribución a la organización, se originan en las relaciones que establecen los individuos con la organización y con otros individuos al ejecutar su trabajo como los relacionados con el trabajo en equipo (comunicación, coordinación, balance, cooperación, tamaño, distribución, cohesión), motivación y satisfacción, sentido de pertenencia y nivel de propiedad del código, por ejemplo.

TABLA NO. 18 CLASIFICACIÓN DE LOS FACTORES HUMANOS

Categoría	Factor humano
Individuales	Competencias Experiencia Profesionalismo Empoderamiento Conciencia de calidad Creatividad Personalidad Carácter Estado de ánimo
Organizacionales	Estructura organizativa Cultura organizacional Estilo de liderazgo Gestión del cambio Gestión del conocimiento
Involucramiento en el trabajo	Trabajo en equipo Motivación Satisfacción Sentido de pertenencia Propiedad del código Entorno ambiental

Nota. Elaboración propia. Se listan únicamente los factores humanos que se identificaron en la revisión de la literatura. Elaboración propia, principalmente a partir de información en Getting the best out of people: ISO 10018 aids ISO 9001 implementation [34], y otras fuentes detalladas en el artículo.

Las categorías identificadas fueron determinadas a partir del análisis de clasificaciones de factores humanos en el campo de la calidad, la aviación, y la salud y seguridad ocupacional.

3.3 Buenas prácticas de los factores humanos

En la revisión de la literatura se identificaron también buenas prácticas relacionadas con los factores humanos identificados, y que según los estudios posibilitan el desarrollo de software de calidad. Las buenas prácticas fueron organizadas en las 3 categorías correspondientes a la clasificación de factores humanos.

El inventario de buenas prácticas identificadas es el siguiente:

3.3.1 Del individuo

- 1) Tener las competencias necesarias de conformidad con los requisitos del trabajo.
 - a. Saber cómo programar, esto es, conocer de lenguajes, bases de datos, sistema operativo, software de control de versiones, revisores de sintaxis, herramientas para pruebas automatizadas, etc.
 - b. Tener un plan de carrera o de crecimiento profesional.
 - c. Buscar los medios para acceder a actividades de formación o capacitación.
 - d. Aprender constantemente nuevos lenguajes, tecnologías, técnicas o herramientas.
 - e. Leer libros técnicos y no técnicos.

- f. Participar en grupos de discusión, sociedades profesionales, conferencias, etc.
- g. Hacer presentaciones, escribir y publicar artículos.
- 2) Ayudar a la organización a promover el crecimiento profesional y el aprendizaje.
 - a. Tomar la iniciativa en establecer un ambiente de aprendizaje.
 - b. Animar a otros a que lo hagan.
- 3) Comportarse como profesional y ayudar a la organización a promover el profesionalismo.
 - a. Conocer su área.
 - b. Hacer lo correcto, decir la verdad, ser honesto.
 - c. Tratar a los demás con respeto.
 - d. Ser un líder.
 - e. No hablar mal de los otros desarrolladores y procesos existentes.
 - f. Servir al cliente con pasión y entrega.
 - g. Ser un solucionador de problemas.
 - h. Ser comprometido.
 - i. Buscar y defender el uso de buenas prácticas de calidad.
- 4) Colaborar o participar en las mejoras de calidad y ayudar a la organización a promover la conciencia de calidad.
 - a. Reconocer y proponer técnicas que beneficiarán a la organización.
 - b. Saber cuánto esfuerzo invertir en las iniciativas de calidad.
 - c. Entender cuando y como hacer cumplir los procesos.
 - d. Cooperar o participar activamente en las revisiones, inspecciones, etc.
- 5) Ayudar a la organización a promover la creatividad y la innovación.
- 6) Ayudar a la organización a promover una estructura organizativa y cultura organizacional apropiada.

3.3.2 De la organización

- 1) Determinar las competencias necesarias de conformidad con los requisitos del trabajo.
 - a. Identificar y priorizar las necesidades de capacitación.
 - b. Desarrollar planes de formación o capacitación continua.
- 2) Proporcionar formación o tomar otras acciones para lograr las competencias necesarias.
 - a. Entrenar en las habilidades nuevas que el trabajo necesita.
 - b. Dar tiempo a los empleados para su participar en actividades de formación o capacitación.
- 3) Evaluar la eficacia de las acciones tomadas para lograr las competencias necesarias.
- 4) Mantener los registros apropiados de la educación, formación, habilidades y experiencia de cada empleado en la organización.
- 5) Seleccionar el personal apropiado para el puesto de trabajo.
 - a. Evaluar las competencias, personalidad, carácter, etc.
 - b. Considerar las competencias y el espíritu de aprendizaje más que los años de experiencia.
- 6) Promover el crecimiento profesional y el aprendizaje.
 - a. Promover la educación constante para mantener los conocimientos y habilidades al día, el aprendizaje de nuevos lenguajes, tecnologías, técnicas o herramientas.
 - b. Promover la lectura de libros técnicos y no técnicos.
 - c. Promover la certificación o acreditación de los empleados.
 - d. Promover la participación en grupos de discusión, sociedades profesionales, conferencias, etc.
 - e. Promover el hacer presentaciones, escribir y publicar artículos.
- 7) Promover el comportamiento profesional.
 - a. Fomentar la iniciativa y confianza del empleado para tomar nuevas responsabilidades.

- b. Fomentar la libertad para actuar y resolver problemas.
 - c. Reconocer los valores y comportamientos que fomentan buenos desarrolladores.
- 8) Incentivar a los empleados.
- a. Que cuiden de su crecimiento profesional.
 - b. Que participen en actividades de formación o capacitación.
 - c. Que promuevan la conciencia de calidad.
 - d. Que tengan buenos resultados.
 - e. Que tengan una actitud investigadora.
 - f. Que implementen nuevas soluciones aprendidas.
 - g. Que compartan su conocimiento con los compañeros de trabajo.
- 9) Proveer acceso a la información necesaria para desempeñar el trabajo.
- a. A información para la toma de decisiones.
 - b. A Internet o correo electrónico para la búsqueda de información que le aporte conocimiento y formas de resolver problemas, o para acceder a redes profesionales o grupos de discusión.
 - c. A bibliografía técnica como revistas, libros, bases de datos de conocimiento, etc.
- 10) Promover la conciencia de calidad.
- a. Identificar las buenas prácticas que puedan ser útiles para la organización.
 - b. Identificar incentivos y reconocimientos que serán efectivos para mejorar los procesos.
 - c. Asegurar que los empleados son conscientes de la importancia de sus actividades y su contribución a los objetivos de calidad, que sientan que el trabajo de cada individuo es notado y que es importante.
- 11) Establecer procedimientos definidos para identificar mejoras de calidad, asignar, programar, completar y verificar el trabajo (como los identificados en la ISO 9001, CMMI o PCMM, etc.)
- a. Reconocer y establecer las buenas prácticas de calidad que deberán observarse y como reconocerlas.
 - b. Reconocer y establecer las técnicas que beneficiarán a la organización.
 - c. Reconocer y establecer un plan de mejora de los procesos de mejora de la calidad.
 - d. Desarrollar planes de pruebas, inspecciones, revisiones, etc.
 - e. No tomar modelos de procesos como un credo, si la organización realmente necesita CMMI, ISO 9001 o PCMM abrir el camino, pero si no, recomendar o implementar pequeñas mejoras graduales.
- 12) Establecer los objetivos de calidad explícitamente.
- a. Objetivos para los atributos externos como la corrección, usabilidad, eficiencia, confiabilidad, adaptabilidad, exactitud y robustez.
 - b. Objetivos para los atributos internos como la mantenibilidad, flexibilidad, portabilidad, reusabilidad, legibilidad, facilidad de ser probado y comprensibilidad.
- 13) Promover la creatividad y la innovación:
- a. Fomentar la independencia.
 - b. Tomar en cuenta los diferentes puntos de vista del problema y no solo lo que uno espera ver (estereotipos).
 - c. Dejar de lado tabúes y no excluir totalmente la intuición o los juicios cualitativos.
 - d. Fomentar la cooperación y el proponer ideas.
 - e. Fomentar la confianza.
 - f. Fomentar un clima con cierto grado de tolerancia a los errores o fallas.
 - g. Fomentar un clima en el que sea posible relajarse.
- 14) Procurar una estructura organizacional propicia para fomentar la calidad.
- a. Que tenga una visión y misión claras y las de a conocer a todos los empleados proveyendo los medios necesarios.
 - b. Que tenga líneas de autoridad y responsabilidad definidas.
 - c. Que no sea compleja al punto de entorpecer la comunicación.

- d. Que tenga procedimientos definidos.
 - e. Que supervise que los procesos se lleven con eficacia y tome las medidas sobre aspectos que limiten las operaciones.
- 15) Procurar una cultura organizacional propicia para fomentar la calidad.
- a. Que promueva actuar según el plan y no la necesidad.
 - b. Que no promueva el negociar soluciones recurriendo a favores.
 - c. Que no promueva el someterse a presiones, requerimientos antojadizos o realizar un mal trabajo por sobre aquellos que se consideran importantes para el éxito del proyecto.
 - d. Que gestione y promueva compartir el conocimiento, ideas y prácticas con los compañeros de trabajo.
 - e. Que provea una atmosfera de confianza y respeto mutuo para expresar ideas y opiniones.
 - f. Que de orientación suficiente para realizar las tareas.
 - g. Que promueva la participación del cliente en el proceso de desarrollo de software.
 - h. Que comparta la visión del producto final con el cliente.
 - i. Que promueva la idea de la calidad como primera prioridad y de la productividad como su consecuencia natural.
 - j. Que promueva la mejora continua del proceso de desarrollo de software.
 - k. Que promueva la detección interna de errores y no por parte del cliente.
 - l. Que cuente con procedimientos formales de desarrollo de software.
 - m. Que cuente con un proceso de desarrollo de software iterativo.
 - n. Que promueva el hacer lo que tiene sentido y no dar lugar a dogmas.
 - o. Que gestione apropiadamente el cambio.
- 16) Promover un estilo de liderazgo propicio para fomentar la calidad.
- a. Que ayude a los empleados a hacer mejor el trabajo y a aprender por métodos objetivos.
 - b. Que sea de tipo colaborativo o directivo según la necesidad.
 - c. Que cuente con líderes involucrados, trabajadores, aptos, preparados para la tarea con anticipación, maximicen el valor de todos los miembros, y hagan sus tareas con buen humor y buena voluntad.

3.3.3 Del involucramiento en el trabajo

- 1) Procurar equipos de trabajo que propicien la calidad.
 - a. Que tengan responsabilidades y habilidades balanceadas entre los miembros.
 - b. Que compartan las metas y objetivos, estén motivados por las metas comunes.
 - c. Que tengan espíritu de grupo y cooperen entre ellos, compartan el conocimiento y se den soporte mutuo.
 - d. Tengan vínculos emocionales y sociales.
 - e. Sean leales y protejan al grupo como entidad.
 - f. Establezcan sus propios estándares y los respeten.
 - g. Promuevan la mejora continua.
 - h. No cuenten con más de 10 miembros.
 - i. Sean co-ubicados o distribuidos.
 - j. Sean propietarios del código.
- 2) Promover un ambiente de trabajo que:
 - a. Tenga una baja rotación de personal.
 - b. Provea un salario y beneficios justos.
 - c. De a los empleados tiempo para socializar.
 - d. Provea los medios que facilite las comunicaciones (como correo electrónico y redes sociales).

- e. De incentivos o reconocimientos por los logros, demuestre a los empleados que son valorados y que su trabajo es importante.
- f. Valore la calidad del producto y no acepte productos mediocres.
- g. Asigne responsabilidades, tareas demandantes pero no imposibles.
- h. Les haga sentir que son parte de la organización y del equipo de trabajo.
- i. Ambiente físico dedicado y privado, ruido aceptable y pocas interrupciones.

3.4 Indicadores para la evaluación de los factores humanos

Se ha identificado una lista de indicadores organizados según las categorías identificadas de factores humanos. Se formulan los indicadores en la forma preguntas para cada una de las buenas prácticas identificadas. Las preguntas se responden con un sí o no se cumple. Pueden ser utilizados para evaluar los factores humanos que inciden en la calidad del software en equipos de trabajo como parte de los procesos de control y aseguramiento de la calidad.

El inventario de indicadores o parámetros es el siguiente:

3.4.1 Del individuo

Los siguientes indicadores deben ser evaluados por alguien que conozca al individuo y el trabajo del individuo, pero no por el individuo mismo:

- ¿Tiene el nivel educativo o las competencias apropiadas para el puesto?
- ¿Ha participado en el último año en capacitaciones o actividades de formación profesional y técnica? ¿Cuál o cuáles?
- ¿Ha aprendido en el último año algún nuevo lenguaje, tecnología técnica o herramienta? ¿Cuál o cuáles?
- ¿Ha experimentado con alguna nueva solución en el último año?
- ¿Ha leído en el último año algún libro técnico o no técnico? ¿Cuál o cuáles?
- ¿Participa en algún grupo de discusión? ¿Cuál o cuáles?
- ¿Participa en alguna sociedad profesional? ¿Cuál o cuáles?
- ¿Ha asistido en el último año a alguna conferencia? ¿Cuál o cuáles?
- ¿Está suscrito a una revista técnica? ¿Cuál o cuáles?
- ¿Ha hecho en el último año alguna presentación dentro o fuera de la empresa? ¿Cuál o cuáles?
- ¿Ha escrito en el último año artículos o ha investigado?
- ¿Ayuda a identificar y priorizar las necesidades de capacitación?
- ¿Anima a los demás a aprender?
- ¿Hace lo correcto, dice la verdad, es honesto?
- ¿Trata a los demás con respeto?
- ¿Es un líder?
- ¿No habla mal de los desarrolladores?
- ¿No habla mal de los procesos existentes?
- ¿Sirve al cliente con pasión y entrega?
- ¿Es un solucionador de problemas?
- ¿Es comprometido?
- ¿Busca y defiende el uso de buenas prácticas de calidad?
- ¿Ayuda a fomentar la iniciativa y confianza del empleado para tomar nuevas responsabilidades?
- ¿Ayuda a fomentar la libertad para actuar y resolver problemas?
- ¿Ayuda a reconocer los valores y comportamientos que fomentan buenos desarrolladores?
- ¿Colabora o participa en las mejoras de calidad?
- ¿Sabe cuánto esfuerzo invertir en las iniciativas de calidad?
- ¿Entiende cuando y como hacer cumplir los procesos?
- ¿Coopera o participa activamente en las inspecciones, revisiones, etc.?

- ¿Ayuda a reconocer y establecer buenas prácticas de calidad?
- ¿Ayuda a reconocer y establecer técnicas que beneficiarán a la organización?
- ¿Ayuda a reconocer y establecer planes de mejora de la calidad?
- ¿Ayuda a establecer los objetivos de calidad para los atributos externos e internos de la calidad y cómo medirlos?
- ¿Ayuda a la organización a promover la creatividad y la innovación?

3.4.2 De la organización

Los siguientes indicadores deben ser verificados por los individuos que trabajan en la organización evaluada:

- ¿La organización identifica y prioriza las necesidades de capacitación de sus empleados?
- ¿La organización cuenta con planes de capacitación o formación continua?
- ¿La organización provee formación para sus empleados?
- ¿La organización entrena en las habilidades nuevas que el trabajo necesita?
- ¿La organización promueve la certificación o acreditación de sus empleados y provee los medios para lograrlo?
- ¿La organización da tiempo a los empleados para su participar en actividades de formación o capacitación?
- ¿Se ha experimentado en su organización con alguna nueva solución en el último año?
- ¿Su organización está suscrita a una revista técnica? ¿Cuál o cuáles?
- ¿Cuenta su organización con una biblioteca o acceso a una base de datos de conocimiento? ¿Cuál o cuáles?
- ¿Evalúan en su organización la eficacia de las acciones tomadas para lograr las competencias necesarias?
- ¿Mantiene su organización los registros apropiados de la educación, formación, habilidades y experiencia de cada empleado en la organización?
- ¿Promueve su organización un ambiente de aprendizaje?
- ¿Promueve su organización la participación en grupos de discusión?
- ¿Promueve su organización la participación en sociedades profesionales?
- ¿Promueve su organización la asistencia y/ participación en conferencias?
- ¿Promueve su organización el hacer presentaciones dentro o fuera de la empresa?
- ¿Promueve su organización la producción de artículos y publicación?
- ¿Promueve su organización que cada individuo conozca su área?
- ¿Promueve su organización el hacer lo correcto, decir la verdad, ser honesto?
- ¿Promueve su organización el tratar a los demás con respeto?
- ¿Promueve su organización el ser un líder?
- ¿Promueve su organización no hablar mal de los otros desarrolladores y procesos existentes?
- ¿Promueve su organización el servir al cliente con pasión y entrega?
- ¿Promueve su organización ser un solucionador de problemas?
- ¿Promueve su organización ser comprometido?
- ¿Promueve su organización el buscar y defender el uso de buenas prácticas de calidad?
- ¿Fomenta su organización la iniciativa y confianza del empleado para tomar nuevas responsabilidades?
- ¿Fomenta su organización la libertad para actuar y resolver problemas?
- ¿Reconoce su organización los valores y comportamientos que fomentan buenos desarrolladores?
- ¿Provee su organización la información suficiente para la toma de decisiones?
- ¿Provee su organización acceso a Internet para búsqueda de información que le aporte conocimiento y formas de resolver problemas que enfrente en su trabajo?
- ¿Provee su organización acceso a Internet u otra red profesional para participar en grupos de discusión?

- ¿Provee su organización acceso a bibliografía técnica como revistas, libros, bases de datos de conocimiento, etc.?
- ¿Identifica su organización las buenas prácticas que puedan ser útiles para la organización?
- ¿Identifica su organización incentivos y reconocimientos que serán efectivos para mejorar los procesos?
- ¿Asegura su organización que los empleados son conscientes de la importancia de sus actividades y su contribución a los objetivos de calidad? ¿Por qué medios?
- ¿Incentiva su organización a los empleados que cuidan de su crecimiento profesional?
- ¿Incentiva su organización a los empleados que participan en actividades de formación o capacitación?
- ¿Incentiva su organización a los empleados que promueven la conciencia de calidad?
- ¿Incentiva su organización a los empleados que tienen buenos resultados?
- ¿Incentiva su organización a los empleados que tienen una actitud investigadora?
- ¿Incentiva su organización a los empleados que implementan nuevas soluciones aprendidas?
- ¿Incentiva su organización a los empleados que comparten su conocimiento con los compañeros de trabajo?
- ¿Reconoce y establece su organización las buenas prácticas de calidad que deberán observarse y como reconocerlas?
- ¿Reconoce y establece su organización las técnicas que beneficiarán a la organización?
- ¿Reconoce y establece su organización planes de mejora de la calidad?
- ¿Desarrolla su organización planes de pruebas, inspecciones, revisiones, etc.?
- ¿Promueve su organización no tomar modelos de procesos como un credo, recomienda o implementa mejoras según la necesidad?
- ¿Establece su organización objetivos para los atributos externos como la corrección, usabilidad, eficiencia, confiabilidad, adaptabilidad, exactitud y robustez, y cómo medirlos?
- ¿Establece su organización objetivos para los atributos internos como la mantenibilidad, flexibilidad, portabilidad, reusabilidad, legibilidad, facilidad de ser probado y comprensibilidad, y cómo medirlos?
- ¿Fomenta su organización la independencia?
- ¿Fomenta su organización la cooperación y el proponer ideas?
- ¿Fomenta su organización la confianza?
- ¿Fomenta su organización un clima con cierto grado de tolerancia a los errores o fallas?
- ¿Fomentar su organización un clima en el que sea posible relajarse?
- ¿Evalúa su organización a los empleados por sus competencias, personalidad, carácter, etc.?
- ¿Considera su organización las competencias y el espíritu de aprendizaje más que los años de experiencia?
- ¿Tiene su organización una visión y misión claras?
- ¿Da a conocer su organización su misión y visión?
- ¿Tiene su organización líneas de autoridad y responsabilidad definidas?
- ¿La estructura organizacional no entorpece la comunicación?
- ¿Tiene su organización procedimientos definidos?
- ¿Supervisa su organización que los procesos se lleven con eficacia y toma las medidas oportunas sobre aspectos que limiten las operaciones?
- ¿Promueve su organización el actuar según el plan y no la necesidad?
- ¿Promueve su organización no negociar soluciones recurriendo a favores?
- ¿Promueve su organización no someterse a presiones, requerimientos antojadizos o realizar un mal trabajo por sobre aquellos que se consideran importantes para el éxito del proyecto?
- ¿Promueve su organización el compartir el conocimiento con otros, compartir ideas y prácticas con los compañeros de trabajo?
- ¿Promueve su organización el confianza para expresar ideas y opiniones, atmosfera de confianza y respeto mutuo?
- ¿Da su organización orientación suficiente para realizar las tareas?

- ¿Promueve su organización la participación del cliente en el proceso de desarrollo de software?
- ¿Promueve su organización el compartir la visión del producto final con el cliente?
- ¿Promueve su organización la idea de la calidad como primera prioridad y de la productividad como su consecuencia natural?
- ¿Promueve su organización la mejora continua del proceso de desarrollo de software?
- ¿Promueve su organización la educación constante para mantener los conocimientos y habilidades al día?
- ¿Promueve su organización la detección interna de errores y no por parte del cliente?
- ¿Promueve su organización la existencia de procedimientos formales de desarrollo de software?
- ¿Promueve su organización un proceso de desarrollo iterativo en todas las etapas del desarrollo de software?
- ¿Promueve su organización el sentimiento de que el trabajo de cada individuo es notado y que es importante?
- ¿Promueve su organización el hacer lo que tiene sentido y no dar lugar a dogmas?
- ¿Se gestiona en su organización el cambio?
- ¿Se gestiona en su organización el conocimiento?
- ¿El líder ayuda a los empleados a hacer mejor el trabajo y a aprender por métodos objetivos?
- ¿El líder es de tipo colaborativo o directivo según la necesidad?
- ¿El líder es involucrado, trabajador, apto, preparado para la tarea con anticipación, maximiza el valor de todos los miembros, y hace sus tareas con buen humor y buena voluntad?

3.4.3 Del involucramiento en el trabajo

Los siguientes indicadores deben ser verificados por los individuos que trabajan para la organización evaluada:

- ¿Los miembros del equipo tienen responsabilidades y habilidades balanceadas?
- ¿Las habilidades de cada individuo en el equipo de trabajo se reconocen y se llevan a buen uso?
- ¿Los miembros del equipo tienen espíritu de grupo, cooperan entre ellos, comparten el conocimiento y se dan soporte mutuo?
- ¿Los miembros del equipo comparten las metas y objetivos comunes?
- ¿Los miembros del equipo tienen vínculos emocionales y sociales?
- ¿Los miembros del equipo tienen son leales y protegen al grupo como entidad?
- ¿Los miembros del equipo tienen establecen sus propios estándares y los respetan?
- ¿Los miembros del equipo promueven la mejora continua?
- ¿Los miembros del equipo no son más de 10?
- ¿Los equipos son co-ubicados o distribuidos?
- ¿Los equipos son propietarios del código que desarrollan?
- ¿Se tiene en su organización una baja rotación de personal?
- ¿Se provee en su organización un salario y beneficios justos?
- ¿Se da en su organización tiempo para socializar?
- ¿Se provee en su organización los medios que facilite las comunicaciones (como correo electrónico y redes sociales)?
- ¿Dan en su organización incentivos o reconocimientos por los logros?
- ¿Se siente usted valorado y que su trabajo es importante?
- ¿Se valora en su organización la calidad del producto y no se acepta productos mediocres?
- ¿Se asignan a usted responsabilidades, tareas demandantes pero no imposibles?
- ¿Se siente usted que es parte de la organización y del equipo de trabajo?
- ¿Tiene usted un ambiente físico dedicado y privado, ruido aceptable y pocas interrupciones?

4 Investigación de campo

La revisión de la literatura reveló que la calidad del software y los factores humanos que inciden en ella es un tema complejo. Es un tema complejo porque la cantidad de factores humanos que pueden incidir en la calidad del software es grande, como muchos y variados son los aspectos que se derivan de los individuos; y además por las interrelaciones que hay entre ellos.

Pese a la complejidad del tema de investigación, la revisión de la literatura identificó y clasificó un conjunto de factores humanos que han sido estudiados y que se ha identificado en contextos particulares que inciden o no en la calidad del software.

Ahora, con la investigación de campo se busca reconocer esos factores humanos que fueron identificados en la revisión de la literatura en el contexto salvadoreño. Se busca comparar estos hallazgos con la opinión y experiencia de profesionales que trabajen o hayan trabajado en el desarrollo de software en El Salvador.

Este reconocimiento se hace en dos fases, la primera que corresponde a la investigación cualitativa busca mediante entrevistas entender el problema y servir de base para la construcción del instrumento de la segunda fase. La segunda fase que corresponde a la investigación cuantitativa busca, haciendo uso de cuestionarios, reconocer y valorar los factores humanos que inciden en la calidad del software con una muestra mayor.

4.1 Fase 1: Investigación cualitativa

4.1.1 Resultados

4.1.1.1 PARTICIPANTES

En la primera fase de la investigación de campo, la investigación cualitativa, se entrevistaron 5 participantes o expertos, con las siguientes características:

TABLA NO. 19 PARTICIPANTES DE LA INVESTIGACIÓN CUANTITATIVA

Características	Descripción
Área profesional	Desarrollo de software.
Ubicación geográfica	El Salvador.
Nivel de estudios	Licenciatura o ingeniería si no tuviera la experiencia.
Años de experiencia	Mínimo 10 años si no tuviera el nivel de estudios indicado arriba o 5 años si lo tuviera.

Nota. Elaboración propia.

4.1.1.2 ANTECEDENTES

La entrevista fue semiestructurada con una duración aproximada de 30 a 45 minutos.

Se caracterizó a cada participante de acuerdo a su edad, nivel educativo, tipo de empresa en la que trabaja, el sector económico de la empresa, su puesto actual, el porcentaje de software que es desarrollado internamente en su empresa y cuanto es comprado, los años de experiencia en el desarrollo de software, los tamaños de equipo y duración de los proyectos en que han trabajado.

Los resultados de la caracterización de los participantes no se detallan (pero pueden verse en el Anexo No. 6) pues el objetivo de esta primera fase es únicamente entender el problema basándose en las opiniones y experiencias de cada uno de los participantes y servir de base para la elaboración del instrumento de investigación de la segunda fase.

4.1.1.3 CONTEXTO DEL PARTICIPANTE

Además de las preguntas para caracterizar a cada participante se indagó sobre el contexto de cada uno de ellos en relación al tema de investigación: que entienden por software de calidad, si evalúan la calidad del software, los aspectos que evalúan, y las herramientas que utilizan.

La mayoría de los participantes definió el “software de calidad” como aquel que:

- 1) No tiene fallas o errores.
- 2) Cumple con los requisitos funcionales especificados por el usuario.

Además indicaron que los atributos o características que definen un software de calidad son principalmente:

- 1) **Corrección** (que cumpla con los requisitos).
- 2) **Eficiencia** (que sea rápido).
- 3) **Disponibilidad** (que esté disponible cuando se necesita).
- 4) **Fiabilidad** (que no tenga fallas o errores).
- 5) **Integridad** (que prevenga el uso no autorizado, incluyendo la confidencialidad).
- 6) **Usabilidad** (que sea fácil de usar).
- 7) **Mantenibilidad** (que cumpla con estándares e implemente mejores prácticas).

A la pregunta de si tenían una unidad o responsable del control y aseguramiento de la calidad del software, la mayoría de los participantes a excepción de uno, respondieron que no tienen una unidad ni responsable. La calidad se evalúa, en la mayoría de los casos, por el mismo desarrollador responsable y no se tiene un procedimiento ni una lista de comprobación formal que guíe la evaluación.

El entrevistado que afirmó contar con una unidad de control y aseguramiento de la calidad en su empresa indicó que, la responsabilidad de esta unidad es conducir las pruebas del software, exclusivamente.

A la pregunta de cuáles son los aspectos de la calidad que evalúan, los participantes coincidieron principalmente en los siguientes:

- 1) **Cumplimiento de los requisitos funcionales.**
- 2) **Cumplimiento de los estándares de desarrollo** (principalmente de nomenclatura).
- 3) **Eficiencia y optimización del código** (exclusivamente del código de acceso a la base de datos).
- 4) **Presencia de fallas o errores.**

Adicionalmente algunos mencionaron los aspectos siguientes:

- 1) Conformidad con los lineamientos de arquitectura.
- 2) Cumplimiento de la normativa legal.
- 3) Identificación de código malicioso.

A la pregunta de cuáles son las técnicas o herramientas utilizadas para evaluar la calidad del software respondieron:

- 1) **Pruebas** (para verificar la presencia de fallas o errores mediante pruebas unitarias, de integración y de aceptación del usuario).
- 2) **Verificación de código** (para verificar el cumplimiento de estándares principalmente).
- 3) **Medición de la eficiencia** (principalmente del código de acceso a la base de datos).

Adicionalmente algunos mencionaron el uso de métricas (las que provee el ambiente de desarrollo y principalmente para el código de acceso a la base de datos, en dos casos; y las que proveen la herramienta de verificación de código SonarQube, en un caso).

4.1.1.4 RECONOCIMIENTO DE LOS FACTORES HUMANOS QUE INCIDEN EN LA CALIDAD DEL SOFTWARE

Los factores, ya sean de origen tecnológico o humano, que según la opinión de los participantes inciden en la calidad del software, son:

- 1) Conocimiento de los requisitos funcionales del software.
- 2) Conocimiento del lenguaje de programación.
- 3) Conocimiento del negocio.
- 4) Conciencia de calidad, compromiso con la calidad.
- 5) Experiencia o conocimiento que se adquiere con los años.
- 6) Valores organizacionales que promuevan la calidad.
- 7) Cultura organizacional que promueva el aprendizaje y la calidad.
- 8) Liderazgo participativo.
- 9) Uso de tecnología reciente que incorpore buenas prácticas.
- 10) Tiempo disponible para el proyecto o el requerimiento.
- 11) Pruebas.
- 12) Estándares de desarrollo.
- 13) Metas por cumplir.
- 14) Ser ordenado, cuidadoso, detallista.
- 15) Ser trabajador, no perder el tiempo.
- 16) Rotación de personal.
- 17) Organización del departamento de desarrollo.
- 18) Asignar tiempo para tareas de refactorización.

A la pregunta de cuales factores, de los indicados en la pregunta anterior, opinaban tenían origen en el recurso humano todos los participantes coincidieron que la mayoría de ellos tienen origen en el recurso humano, y son los siguientes:

- 1) Conocimiento de los requisitos funcionales del software.
- 2) Conocimiento del lenguaje de programación.
- 3) Conocimiento del negocio.
- 4) Conciencia de calidad, compromiso con la calidad.
- 5) Experiencia o conocimiento que se adquiere con los años.
- 6) Valores organizacionales que promuevan la calidad.
- 7) Cultura organizacional que promueva el aprendizaje y la calidad.
- 8) Liderazgo participativo.
- 9) Ser ordenado, cuidadoso, detallista.
- 10) Ser trabajador, no perder el tiempo.
- 11) Rotación de personal.
- 12) Organización del departamento de desarrollo.

4.1.1.5 VALORACIÓN DE LOS FACTORES HUMANOS QUE INCIDEN EN LA CALIDAD DEL SOFTWARE

Los 3 factores humanos más recurrentes que los participantes opinan que inciden en mayor medida en la calidad del software, fueron:

- 1) **Conciencia de calidad.**
- 2) **Conocimiento** del lenguaje de programación, de los requisitos funcionales, del negocio.
- 3) **Cultura y valores organizacionales** que promuevan la calidad.

4.1.2 Análisis y discusión

La noción de software de calidad de los participantes es consistente con las definiciones actuales de calidad en cuanto a su énfasis en la idea de que el software de calidad es aquel que no presenta defectos y que cumple con los requisitos especificados.

Precisamente la definición de calidad de la IEEE [21] dice que la calidad del software es:

“a) El grado en que un sistema, componente o proceso cumple con los requisitos especificados.

b) El grado en que un sistema, componente o proceso cumple con las necesidades o expectativas del cliente o usuario”.

Y la definición del “Padre de la Calidad del Software”, Humphrey Watts [22] dice que el software de calidad es aquel que:

“a) Cumple con las necesidades funcionales del usuario, con la fiabilidad y consistencia de realizar el trabajo del usuario.

b) No tiene defectos (algo que está mal en el programa, en el diseño, en sus requisitos, especificaciones, o en la documentación)”.

En los resultados también es evidente que, aunque se reconoce la importancia de la calidad, todavía no se cuenta en las empresas de los participantes con unidades o responsables dedicados exclusivamente a evaluar la calidad del software y que, en los casos que se hace la evaluación, ésta consiste exclusivamente en:

- 1) **Verificar la existencia de fallas o errores** mediante pruebas del software.
- 2) **Verificar el cumplimiento de los requisitos funcionales** mediante pruebas del software.
- 3) **Verificar el cumplimiento de estándares de desarrollo** mediante la verificación del código.
- 4) **Verificar la eficiencia** exclusivamente del código de acceso a la base de datos.

Y que las técnicas de evaluación consisten principalmente en:

- 1) **Pruebas.**
- 2) **Verificación de código.**

En cuanto a los factores que opinan inciden en la calidad del software, los resultados de las entrevistas coinciden en que la mayoría de ellos tienen su origen en el recurso humano y coinciden con algunos de los factores identificados en la revisión de la literatura, como:

- 1) Competencias.
- 2) Conciencia de calidad.
- 3) Experiencia.
- 4) Estructura organizativa.
- 5) Cultura organizacional.
- 6) Estilo de liderazgo.
- 7) Satisfacción.
- 8) Carácter.
- 9) Personalidad.

4.2 Fase 2: Investigación cuantitativa

4.2.1 Resultados

4.2.1.1 PARTICIPANTES

En la segunda fase de la investigación de campo, la investigación cuantitativa se encuestaron 40 participantes o expertos de 20 empresas salvadoreñas, con las siguientes características:

TABLA NO. 20 PARTICIPANTES DE LA INVESTIGACIÓN CUANTITATIVA

Características	Descripción
Área profesional	Desarrollo de software.
Ubicación geográfica	El Salvador.
Nivel de estudios	Cualquier nivel.
Años de experiencia	Cualquiera.

Nota. Elaboración propia.

Originalmente se pensó en la fase cuantitativa excluir participantes con menos de 3 años de experiencia. Sin embargo a partir de la revisión de la literatura, que indica que los años de experiencia no es una característica que se considere tenga impacto en la calidad del software, y a partir de los resultados en las entrevistas, se concluyó que no era prudente pues se excluiría a profesionales recién graduados que inician su carrera, y que tiene desde ya cierta experiencia y podrían aportar su visión a la investigación.

4.2.1.2 ANTECEDENTES

Las encuestas consistieron en un cuestionario de 50 preguntas cerradas, auto administrado, en línea utilizando el software Google Forms, y con una duración aproximada de 15 a 20 minutos.

El cuestionario categorizaba los factores humanos según las categorías identificadas y que fueron determinadas a partir del análisis de clasificaciones de factores humanos en el campo de la calidad, la aviación, y la salud y seguridad ocupacional.

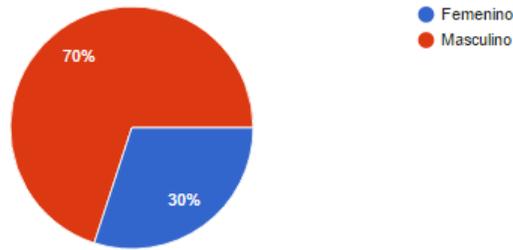
Los factores humanos que se consideraron en el cuestionario no incluyó un factor mencionado en la revisión de la literatura por no haber identificado estudios concluyentes sobre su incidencia en la calidad del software y porque no fue mencionado en la primera fase de la investigación de campo por los profesionales participantes, este factor es la *Gestión del cambio*.

Las primeras preguntas buscaban caracterizar a cada participante de acuerdo a su edad, nivel educativo, tipo de empresa en la que trabaja, el sector económico de la empresa, su puesto actual, los años de experiencia en el desarrollo de software, los tamaños de equipo y duración de los proyectos en que han trabajado.

En total se entrevistaron 40 profesionales distribuidos en aproximadamente 20 empresas de El Salvador.

De los 40 profesionales participantes, 70% se identificaron como hombres y 30% como mujeres.

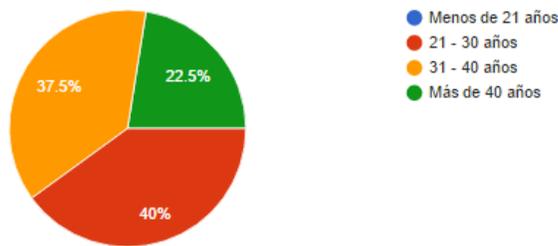
FIGURA NO. 12 DISTRIBUCIÓN POR SEXO



Nota. Elaboración propia.

El 40% de los participantes dijo tener entre 21-30 años, el 37.5% entre 31-40 años y el 22.2% más de 40 años.

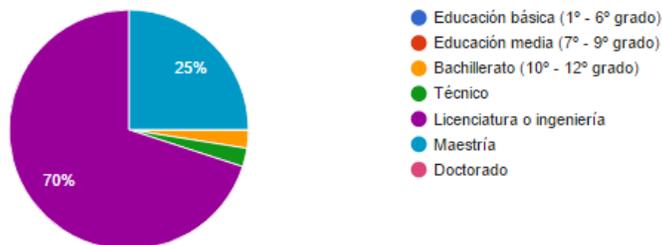
FIGURA NO. 13 DISTRIBUCIÓN POR EDAD



Nota. Elaboración propia.

El nivel educativo de la mayoría de los participantes fue el de Licenciatura o Ingeniería (70%), seguido de participantes con grado de Maestría (25%). El resto de participantes dijo tener un grado de Bachillerato o Técnico.

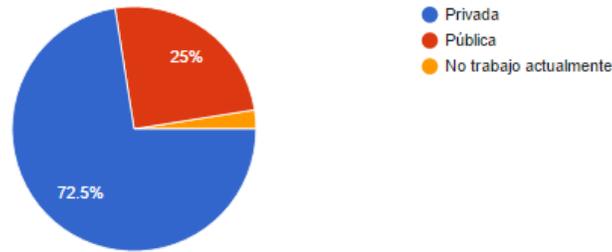
FIGURA NO. 14 DISTRIBUCIÓN POR NIVEL EDUCATIVO



Nota. Elaboración propia.

El 72.5% de los participantes dijo trabajar en la empresa privada, y el 25% en la empresa pública. Uno de ellos dijo no trabajar actualmente.

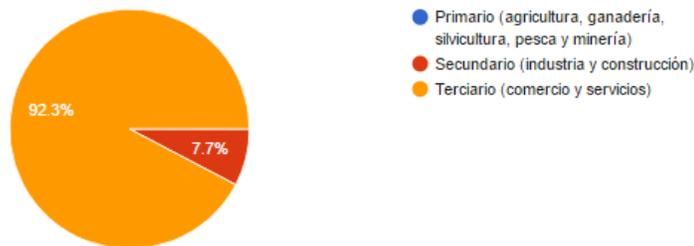
FIGURA NO. 15 DISTRIBUCIÓN POR TIPO DE EMPRESA



Nota. Elaboración propia.

De los participantes que actualmente trabajan, la mayoría dijo trabajar en una empresa dedicada al comercio y los servicios (92.3%).

FIGURA NO. 16 DISTRIBUCIÓN POR SECTOR ECONÓMICO



Nota. Elaboración propia.

De los 40 participantes, 33 de ellos trabajan en empresas grandes de más de 100 empleados,

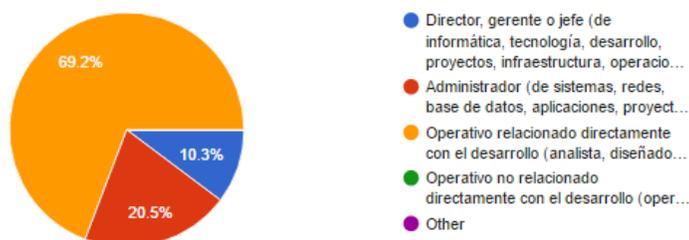
TABLA NO. 21 DISTRIBUCIÓN POR TAMAÑO DE EMPRESA

Tamaño	Número de empresas
Micro (menos de 5 empleados)	2
Pequeña (5 - 49 empleados)	3
Mediana (50 - 99 empleados)	1
Grande (100 o más empleados)	33

Nota. Elaboración propia.

Más del 60% de los participantes en la investigación trabajan en un puesto operativo relacionado directamente con el desarrollo de software. Aproximadamente el 30% en puestos administrativos, y aproximadamente el 10% en puestos directivos, de gerencia o jefaturas.

FIGURA NO. 17 DISTRIBUCIÓN POR PUESTO DE TRABAJO

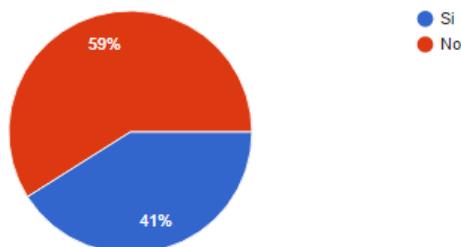


Nota. Elaboración propia.

4.2.1.3 CONTEXTO DEL PARTICIPANTE

El 59% de los participantes dijo que su empresa no cuenta con una unidad o responsable del aseguramiento de la calidad.

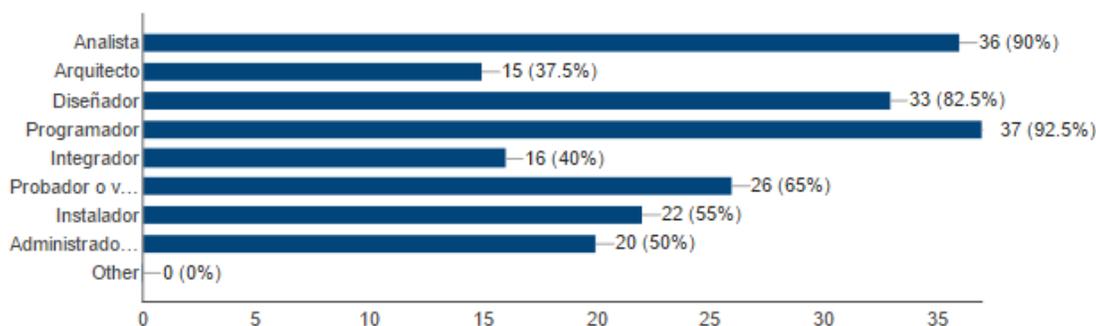
FIGURA NO. 18 EXISTENCIA DE UNIDAD O RESPONSABLE DE CONTROL O ASEGURAMIENTO DE LA CALIDAD



Nota. Elaboración propia.

Los roles en el desarrollo de software que más han desempeñado o desempeñan los participantes de la investigación son los de analista, programador, diseñador y probador o verificador de software.

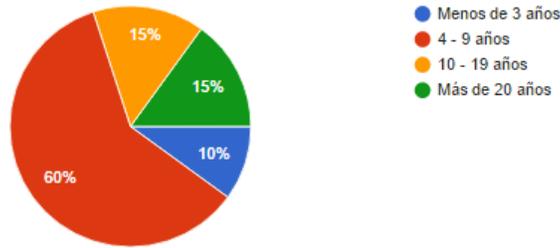
FIGURA NO. 19 ROLES DESEMPEÑADOS EN EL DESARROLLO DE SOFTWARE



Nota. Elaboración propia.

El 60% de los participantes en la investigación dijeron tener entre 4 y 9 años de experiencia en el desarrollo de software, un 15% entre 10 y 19 años de experiencia, y un 15% más de 20 años de experiencia..

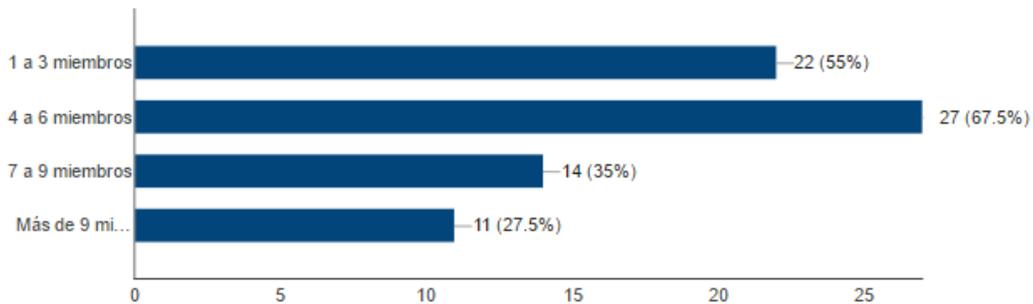
FIGURA NO. 20 DISTRIBUCIÓN POR AÑOS DE EXPERIENCIA



Nota. Elaboración propia.

Los tamaños de equipo de desarrollo más comunes, en los que los participantes de la investigación han trabajado, son equipos de desarrollo pequeños, de 1 a 3 miembros, y de 4 a 6 miembros.

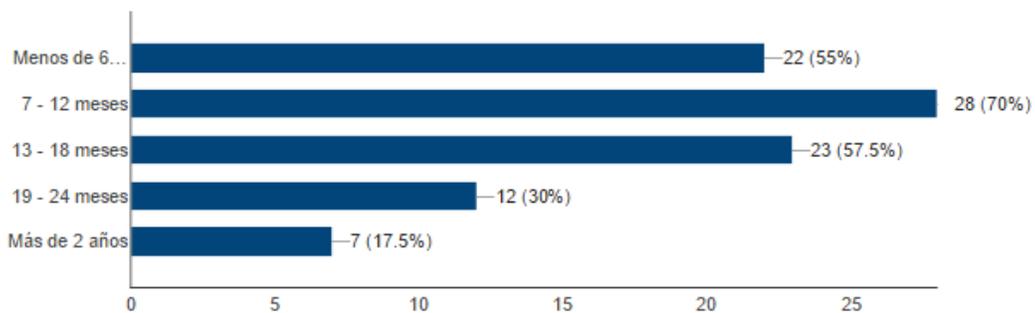
FIGURA NO. 21 TAMAÑOS DE LOS EQUIPOS DE DESARROLLO



Nota. Elaboración propia.

La duración de los proyectos más comunes, en los que los participantes de la investigación han trabajado, son proyectos de menos de año y medio.

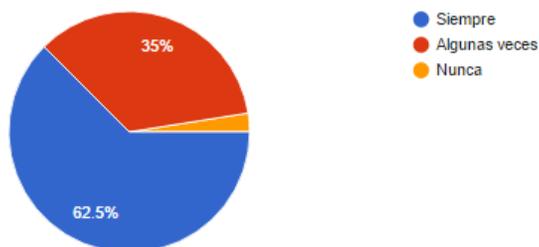
FIGURA NO. 22 DURACIÓN DE LOS PROYECTOS



Nota. Elaboración propia.

La mayoría de los participantes (97.5%) dijo evaluar la calidad del software que desarrolla ya sea "siempre" o "algunas veces".

FIGURA NO. 23 EVALUACIÓN DE LA CALIDAD



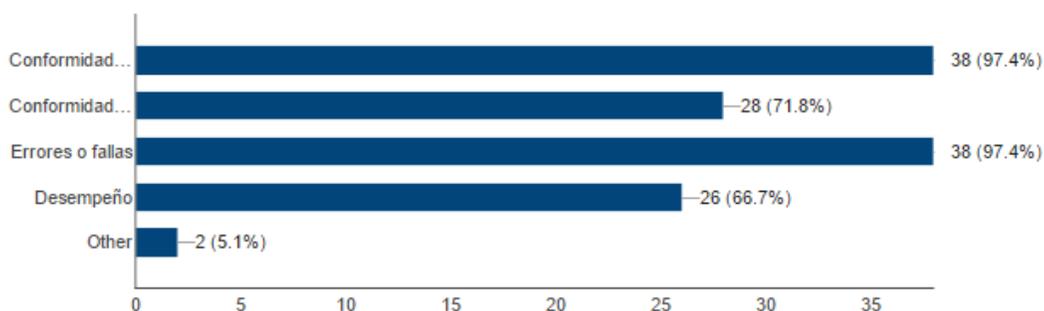
Nota. Elaboración propia.

Quienes evalúan la calidad del software dijeron evaluar principalmente la conformidad con lo requisitos funcionales, y si el software desarrollado presenta errores o fallas.

Con menor frecuencia la conformidad con estándares de desarrollo y el desempeño.

Dos de los participantes dijeron realizar otro tipo de evaluación: las buenas prácticas de desarrollo, y la experiencia de usuario (*User Experience*).

FIGURA NO. 24 ASPECTOS QUE EVALÚA

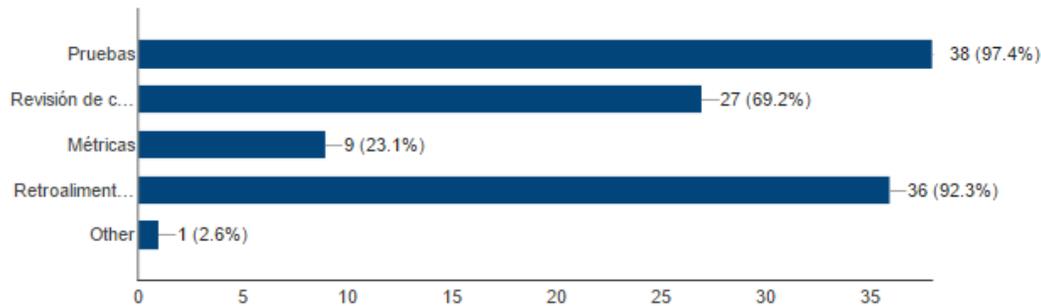


Nota. Elaboración propia.

La mayoría de los participantes dijeron evaluar la calidad del software principalmente mediante las pruebas y la retroalimentación del usuario.

En menor medida mediante la revisión del código, y en mucha menor medida mediante métricas. Uno de los participantes mencionó el uso del software SonarQube.

FIGURA NO. 25 TÉCNICAS DE EVALUACIÓN



Nota. Elaboración propia.

4.2.1.4 RECONOCIMIENTO DE LOS FACTORES HUMANOS QUE INCIDEN EN LA CALIDAD DEL SOFTWARE

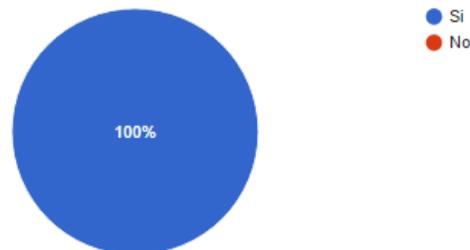
En cuanto a los factores humanos que según la experiencia y la opinión de los participantes inciden en la calidad del software, los resultados fueron:

4.2.1.4.1 COMPETENCIAS

Las competencias se refieren a la capacidad de aplicar o usar un conjunto de conocimientos, destrezas y habilidades relacionadas para realizar con éxito las funciones o tareas de un trabajo.

El 100% de los participantes opina que las competencias inciden en la calidad del software.

FIGURA NO. 26 COMPETENCIAS



Nota. Elaboración propia.

La mayoría coincide que las competencias que inciden positivamente en la calidad del software son: conocimiento técnico, conocimiento del negocio, pensamiento lógico y analítico, habilidad para aplicar conocimiento, habilidad para resolver problemas, habilidad investigativa, habilidad de comunicación, habilidad para tomar decisiones, atención al detalle y la motivación para crear o mejorar artefactos.

Algunos participantes opinan que el conocimiento del negocio, la habilidad para aplicar conocimiento, la habilidad para resolver problemas, la habilidad de comunicación, la habilidad para tomar decisiones y la motivación para crear o mejorar artefactos no incide en la calidad del software.

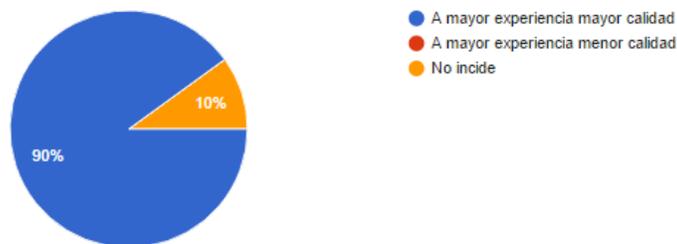
En menor medida algunos participantes opinan que la habilidad de comunicación y la motivación para crear o mejorar artefactos incide negativamente.

4.2.1.4.2 EXPERIENCIA

La experiencia se refiere a la forma de conocimiento o habilidad derivados de la observación, participación y vivencia de un evento.

El 90% de los participantes opina que la experiencia incide positivamente en la calidad del software, mientras que el 10% opina que no incide.

FIGURA NO. 27 EXPERIENCIA



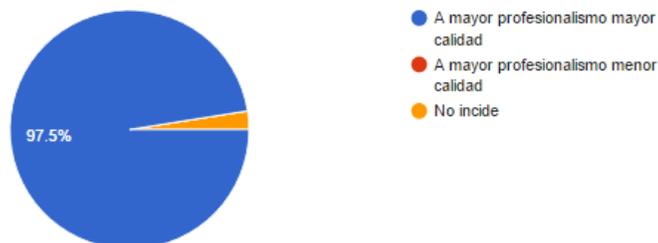
Nota. Elaboración propia.

4.2.1.4.3 PROFESIONALISMO

El profesionalismo se refiere a la característica de la persona que desempeña un trabajo con pericia, aplicación, seriedad, honradez y eficacia.

El 97.5% de los participantes opina que el profesionalismo incide positivamente en la calidad del software, mientras que el 2,5% opina que no incide.

FIGURA NO. 28 PROFESIONALISMO



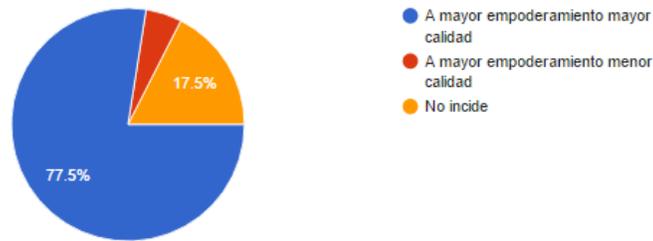
Nota. Elaboración propia.

4.2.1.4.4 EMPODERAMIENTO

El empoderamiento se refiere la práctica de gestión de compartir información, recompensas, y poder con los empleados para que puedan tomar la iniciativa y tomar decisiones para resolver problemas y mejorar el servicio y el rendimiento.

El 77.5% de los participantes opina que el empoderamiento incide positivamente en la calidad del software, mientras que el 17.5% opina que no incide, y el 5% que incide negativamente.

FIGURA NO. 29 EMPODERAMIENTO



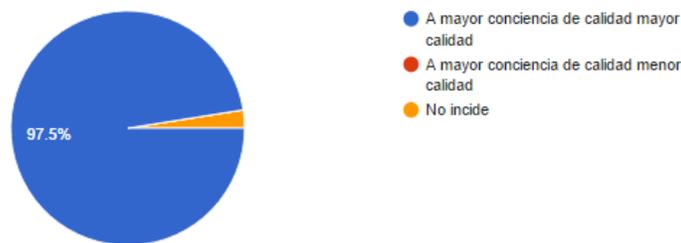
Nota. Elaboración propia.

4.2.1.4.5 CONCIENCIA DE CALIDAD

La conciencia de calidad se refiere a comprender la pertinencia e importancia de las actividades que cada individuo realiza y de cómo estas contribuyen al logro de los objetivos de la calidad.

El 97.5% de los participantes opina que la la conciencia de la calidad incide positivamente en la calidad del software, mientras que el 2.5% opina que no incide.

FIGURA NO. 30 CONCIENCIA DE CALIDAD



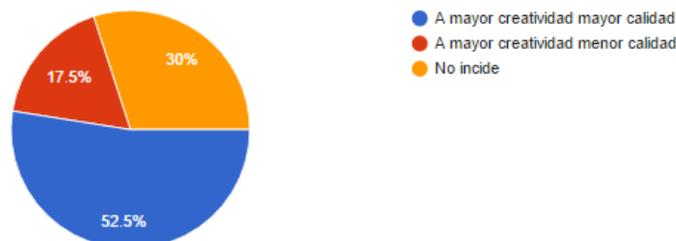
Nota. Elaboración propia.

4.2.1.4.6 CREATIVIDAD

La creatividad se refiere a la capacidad de generar nuevas ideas o conceptos, o de nuevas asociaciones entre ideas y conceptos conocidos, que habitualmente producen soluciones originales. Es sinónimo de pensamiento original, imaginación constructiva, pensamiento divergente o pensamiento creativo.

El 52.5% de los participantes opina que la creatividad incide positivamente en la calidad del software, mientras que el 30% opina que no incide, y el 17.5% que incide negativamente.

FIGURA NO. 31 CREATIVIDAD



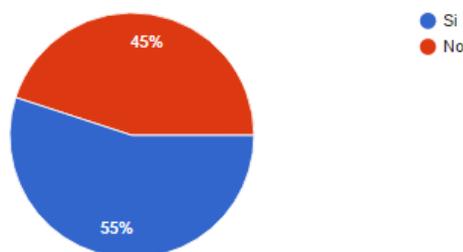
Nota. Elaboración propia.

4.2.1.4.7 PERSONALIDAD

La personalidad se refiere al conjunto de rasgos con las que una persona nace y que pueden ser afectados por su desarrollo social y cultural. Por ejemplo: extrovertido, introvertido, racional, emocional, etc.

El 55% de los participantes opina que la personalidad incide positivamente en la calidad del software, mientras que el 45% opina que no incide.

FIGURA NO. 32 PERSONALIDAD



Nota. Elaboración propia.

Los participantes que opinan que la personalidad incide en la calidad del software valoran los rasgos de personalidad de la siguiente forma:

- 1) **Extrovertido (E)**: aproximadamente igual cantidad de participantes opina que incide positivamente o que no incide.
- 2) **Introvertido (I)**: la mayoría opina que incide negativamente o que no incide.
- 3) **Intuitivo (N)**: todos los participantes opinan que incide positivamente.
- 4) **Sensorial (S)**: aproximadamente igual cantidad de participantes opina que incide positivamente o que no incide.
- 5) **Racional (T)**: casi la totalidad de los participantes opina que incide positivamente.
- 6) **Emocional (F)**: aproximadamente igual cantidad de participantes opina que incide negativamente o no incide.
- 7) **Juzgador (J)**: aproximadamente igual cantidad de participantes opina que incide positivamente o negativamente.
- 8) **Perceptivo (P)**: casi la totalidad de los participantes opina que incide positivamente.

De esto resulta que los perfiles de personalidad que opinan los participantes que inciden positivamente en la calidad del software son:

TABLA NO. 22 ESTILOS DE PERSONALIDAD

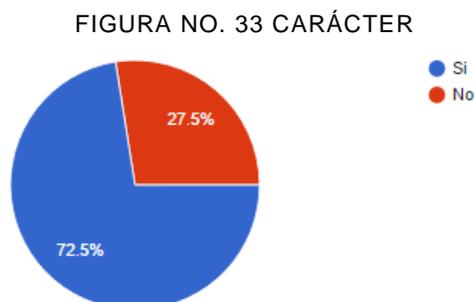
Rasgos	Tipo	Descripción	
Extrovertido, intuitivo, racional, juzgador	ENTJ	Comandante	Líderes audaces, imaginativos y de voluntad fuerte, siempre en busca de un camino, o creando uno.
Extrovertido, intuitivo, racional, perceptivo	ENTP	Innovador	Pensadores inteligentes y curiosos que no pueden resistir un reto intelectual.
Extrovertido, sensorial, racional, juzgador	ESTJ	Ejecutivo	Administradores excelentes, inigualables al administrar cosas o personas.
Extrovertido, sensorial, racional, perceptivo	ESTP	Emprendedor	Personas inteligentes, enérgicas y muy perceptivas, que realmente disfrutan vivir al límite.

Nota. Elaboración propia.

4.2.1.4.8 CARÁCTER

El carácter se refiere al conjunto de rasgos que definen la integridad de la persona, el conjunto de valores que tiene y que son aprendidos de otros (padres, maestros, amigos, etc.) Por ejemplo: responsable, ordenado, honesta, independiente, amigable, etc.

El 72.5% de los participantes opina que el carácter incide positivamente en la calidad del software, mientras que el 27.5% opina que no incide.:



Nota. Elaboración propia.

Los participantes que respondieron que el carácter incide en la calidad del software indican los siguientes rasgos que inciden positivamente:

- 1) Ser apasionado.
- 2) Tener mente abierta.
- 3) Ser sistemático.
- 4) Ser cuidadoso.
- 5) Ser responsable.
- 6) Ser enfocado.
- 7) Ser honesto.
- 8) Ser competitivo.
- 9) Ser flexible.
- 10) Ser seguro.

El rasgo que opinan que incide negativamente en la calidad del software es ser pesimista.

4.2.1.4.9 ESTADO DE ÁNIMO

El estado de ánimo consiste en la actitud o disposición emocional en un momento determinado.

El 77.5% de los participantes opina que el estado de ánimo positivo contribuye a una mayor calidad, mientras que el 22.5% opina que no incide.



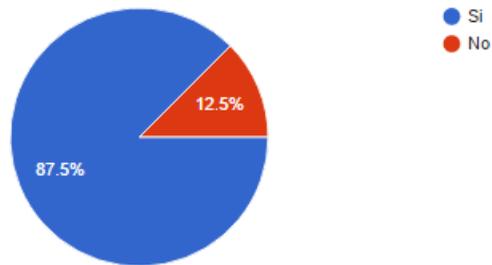
Nota. Elaboración propia.

4.2.1.4.10 ESTRUCTURA ORGANIZATIVA

La estructura organizativa se refiere a la disposición de las líneas de autoridad, comunicaciones, derechos y deberes que determina cómo se asignan los roles, el poder y las responsabilidades; como son controlados y coordinados; y cómo la información fluye entre los diferentes niveles de gestión.

El 87.5% de los participantes opina que la estructura organizativa incide en la calidad del software, mientras que el 12.5% que no.

FIGURA NO. 35 ESTRUCTURA ORGANIZATIVA



Nota. Elaboración propia.

Los participantes que opinan que la estructura organizativa incide en la calidad del software opinan que los aspectos que inciden positivamente son:

- 1) Flexibilidad de la organización.
- 2) Líneas de autoridad claramente definidas.
- 3) Responsabilidades claramente definidas.
- 4) Procedimientos claramente definidos.
- 5) Subdivisión del trabajo por roles.
- 6) Unidad de aseguramiento de la calidad.

La complejidad de la estructura organizativa se considera que incide negativamente.

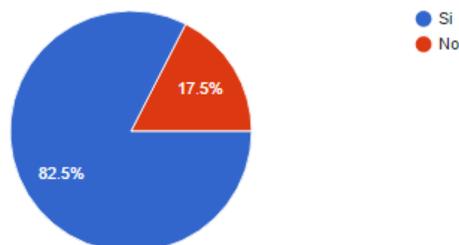
En relación a la existencia de controles estrictos no se tiene una opinión definitiva, unos participantes opinan que incide positivamente mientras que otros que incide negativamente.

4.2.1.4.11 CULTURA ORGANIZACIONAL

La cultura organizacional se refiere a los conocimientos, creencias, valores, costumbres y otros hábitos adquiridos como miembro de la organización.

El 82.5% de los participantes opina que la cultura organizacional incide en la calidad del software, mientras que el 17.5% que no.

FIGURA NO. 36 CULTURA ORGANIZACIONAL



Nota. Elaboración propia.

Los que opinan que la cultura organizacional incide en la calidad del software indican que los aspectos que inciden positivamente son:

- 1) Cultura de planificación.
- 2) Cultura de protección del alcance.
- 3) Cultura de compromiso con la calidad.
- 4) Clima de estabilidad.
- 5) Clima de respeto y confianza.
- 6) Cultura de seguir el proceso debido.
- 7) Cultura de cabildeo (negociar soluciones).
- 8) Cultura de mejora continua.

Entre los aspectos de la cultura organizacional que inciden negativamente se indica la cultura de búsqueda de culpables.

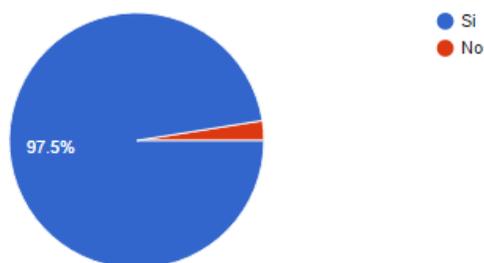
La opinión respecto a la incidencia de la cultura de protección del presupuesto, el plazo y la evaluación por metas está dividida entre los que opinan que incide positivamente y los que opinan que incide negativamente.

4.2.1.4.12 ESTILO DE LIDERAZGO

El estilo de liderazgo es la forma en que el líder dirige a un grupo de personas, proporciona indicaciones, las motiva e implementa planes. Estilos de liderazgo son: colaborativo o participativo, directivo, etc.

El 97.5% de los participantes opina que el estilo de liderazgo incide en la calidad del software, mientras que el 2.5% opina que no incide.

FIGURA NO. 37 ESTILO DE LIDERAZGO



Nota. Elaboración propia.

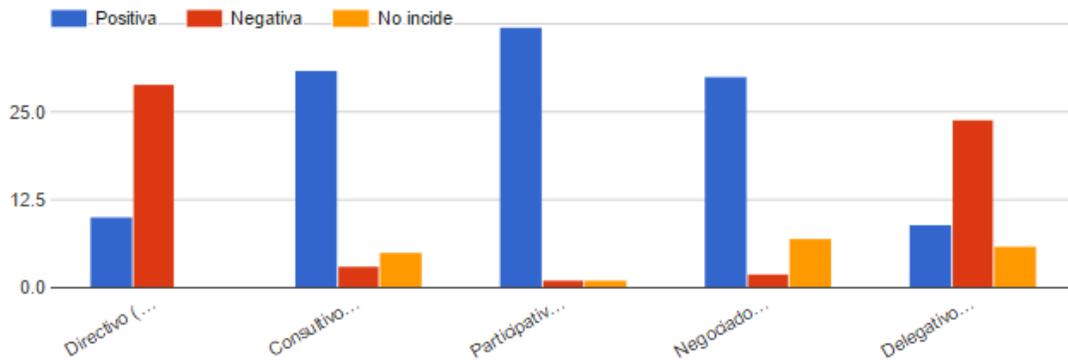
Los estilos de liderazgo que se considera inciden positivamente son:

- 1) **Consultivo** (el líder busca el consejo del equipo antes de tomar una decisión sobre lo que debe hacerse).
- 2) **Participativo** (el líder se ubica como miembro y discute la posible decisión con el equipo).
- 3) **Negociador** (el líder utiliza incentivos para motivar al equipo a hacer ciertas cosas).

Y los estilos de liderazgo que inciden negativamente:

- 1) **Directivo** (el líder dice al equipo que hacer y como hacerlo).
- 2) **Delegativo** (el líder deja que el equipo tome sus propias decisiones sobre lo que debe hacerse).

FIGURA NO. 38 ESTILOS DE LIDERAZGO QUE INCIDEN



Nota. Elaboración propia.

4.2.1.4.13 GESTIÓN DEL CONOCIMIENTO

La gestión del conocimiento se refiere a identificar, usar y compartir de forma eficiente los activos intelectuales de una organización para mejorar su rendimiento y competitividad.

La mayoría de los participantes (95%) opina que la gestión del conocimiento incide positivamente en la calidad del software..

FIGURA NO. 39 GESTIÓN DEL CONOCIMIENTO



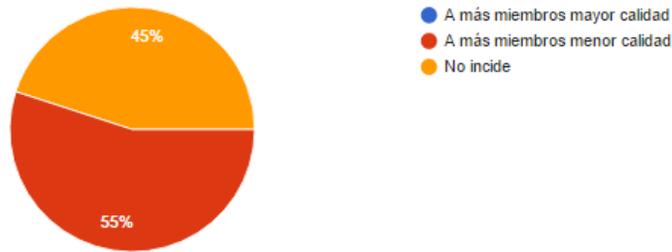
Nota. Elaboración propia.

4.2.1.4.14 TAMAÑO DEL EQUIPO

El tamaño del equipo de desarrollo se refiere al número de miembros en el equipo de desarrollo.

El 55% de los participantes opina que equipos de desarrollo más grandes inciden negativamente en la calidad del software, mientras que el 45% opina que no inciden.

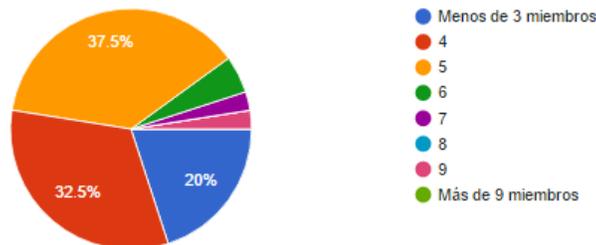
FIGURA NO. 40 TAMAÑO DEL EQUIPO



Nota. Elaboración propia.

La mayoría de los participantes que respondieron que el tamaño del equipo incide en la calidad del software, opinan que el tamaño del equipo óptimo está entre los 4 y 5 miembros. Un 20% de ellos opina que el tamaño óptimo es de menos de 3 miembros.

FIGURA NO. 41 TAMAÑO DE EQUIPO ÓPTIMO



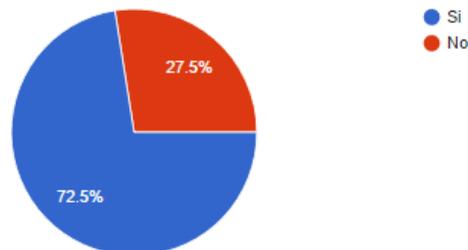
Nota. Elaboración propia.

4.2.1.4.15 DISTRIBUCIÓN DEL EQUIPO

La distribución del equipo de desarrollo se refiere a la localización de los miembros del equipo de desarrollo. Por ejemplo: en el mismo lugar o distribuidos en diferentes lugares (oficinas, departamentos, países, continentes, etc.)

El 72.5% de los participantes opina que la distribución del equipo incide en la calidad del software, mientras que el 27.5% opina que no incide.

FIGURA NO. 42 DISTRIBUCIÓN DEL EQUIPO



Nota. Elaboración propia.

La mayoría de los participantes que respondieron que la distribución del equipo incide en la calidad del software, opinan que los equipos co-ubicados (los miembros trabajan en el mismo lugar) promueven la calidad del software.

Mientras que los equipos globalizados o distribuidos (los miembros trabajan distribuidos geográficamente), y wxternalizados (equipos con responsabilidades diferentes que trabajan fuera de la organización) no la promueven.

FIGURA NO. 43 MODELOS DE DISTRIBUCIÓN DEL EQUIPO



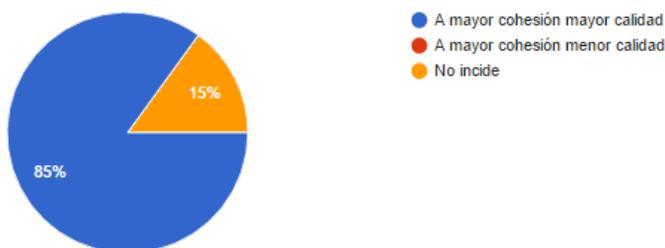
Nota. Elaboración propia.

4.2.1.4.16 COHESIÓN DEL EQUIPO

La cohesión del equipo se refiere al grado en que los miembros del equipo contribuyen a la habilidad del grupo de funcionar como una unidad. Equipos cohesivos tienen lazos emocionales y sociales que los vinculan unos a otros y les motiva por las metas comunes.

La mayoría de los participantes (85%) respondieron que la cohesión del equipo incide positivamente en la calidad del software.

FIGURA NO. 44 COHESIÓN DEL EQUIPO



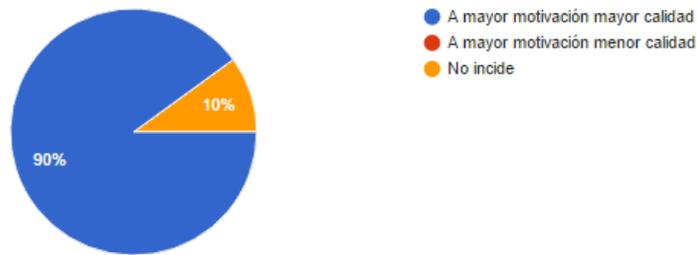
Nota. Elaboración propia.

4.2.1.4.17 MOTIVACIÓN

La motivación se refiere a la condición de estar entusiasmados por actuar o trabajar.

La mayoría de los participantes (90%) respondieron que la motivación incide positivamente en la calidad del software.

FIGURA NO. 45 MOTIVACIÓN



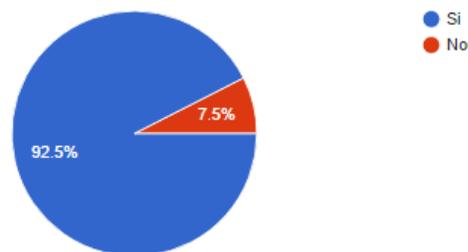
Nota. Elaboración propia.

4.2.1.4.18 SATISFACCIÓN

La satisfacción se refiere al grado en que el individuo siente que sus necesidades (fisiológicas, de seguridad, sociales, de estima y de auto-realización) están satisfechas.

La mayoría de los participantes (92.5%) respondieron que la satisfacción incide positivamente en la calidad del software.

FIGURA NO. 46 SATISFACCIÓN



Nota. Elaboración propia.

Entre los aspectos relacionados con la satisfacción que se indica que inciden positivamente están:

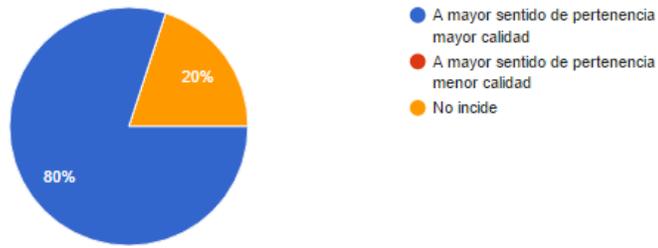
- 1) Tener un salario apropiado.
- 2) Tener un plan de beneficios.
- 3) La formación continua.
- 4) La contratación directa.
- 5) El contar con acceso a Internet.
- 6) El contar con acceso al correo electrónico.
- 7) Los reconocimientos.

4.2.1.4.19 SENTIDO DE PERTENENCIA

El sentido de pertenencia se refiere a la necesidad emocional de ser aceptado como miembro de la organización o del equipo de trabajo.

La mayoría de los participantes (80%) respondieron que el sentido de pertenencia incide positivamente en la calidad del software.

FIGURA NO. 47 SENTIDO DE PERTENENCIA



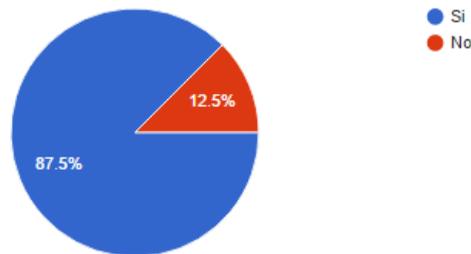
Nota. Elaboración propia.

4.2.1.4.20 PROPIEDAD DEL CÓDIGO

La propiedad del código se refiere a distinguir quién trabajará en qué código, y a cuanto trabajo puede ser o debe ser compartido entre los miembros del equipo o diferentes equipos en la organización. Si hay alguien claramente responsable del código y de su calidad.

La mayoría de los participantes (87.5%) opinan que la propiedad del código incide positivamente en la calidad del software.

FIGURA NO. 48 PROPIEDAD DEL CÓDIGO



Nota. Elaboración propia.

El modelo de propiedad del código que opinan que favorece más la calidad es el modelo débil (el sistema es asignado a un desarrollador, pero se permite la modificación por otros bajo la supervisión del dueño).

Y los que inciden negativamente en la calidad del software son el modelo fuerte (el sistema es propiedad de un solo desarrollador y solo él puede modificarlo), y el colectivo (cualquiera puede hacer modificaciones).

FIGURA NO. 49 MODELOS DE PROPIEDAD DEL CÓDIGO



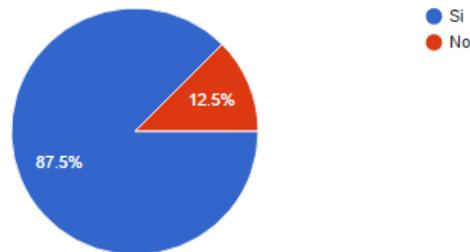
Nota. Elaboración propia.

4.2.1.4.21 ENTORNO AMBIENTAL

El entorno ambiental se refiere a los objetos y condiciones, tangibles y materiales que rodean el ambiente de trabajo.

La mayoría de los participantes (87.5%) opina que el entorno ambiental incide en la calidad del software.

FIGURA NO. 50 ENTORNO AMBIENTAL



Nota. Elaboración propia.

Entre los aspectos del entorno ambiental que los participantes indican que inciden positivamente están:

- 1) Contar con equipo de computación apropiado.
- 2) Contar con amueblamiento apropiado.
- 3) Contar con espacios de trabajo privados.
- 4) Contar con espacios para descanso y socialización.
- 5) Contar con buena iluminación.
- 6) Un bajo nivel de ruido.

El contar con espacios de trabajo abiertos tiene una opinión dividida entre los participantes.

4.2.1.5 OTROS FACTORES HUMANOS QUE INCIDEN EN LA CALIDAD DEL SOFTWARE

Los otros factores humanos que se mencionan inciden en la calidad del software, pueden ser incluidos en los factores identificados en la investigación:

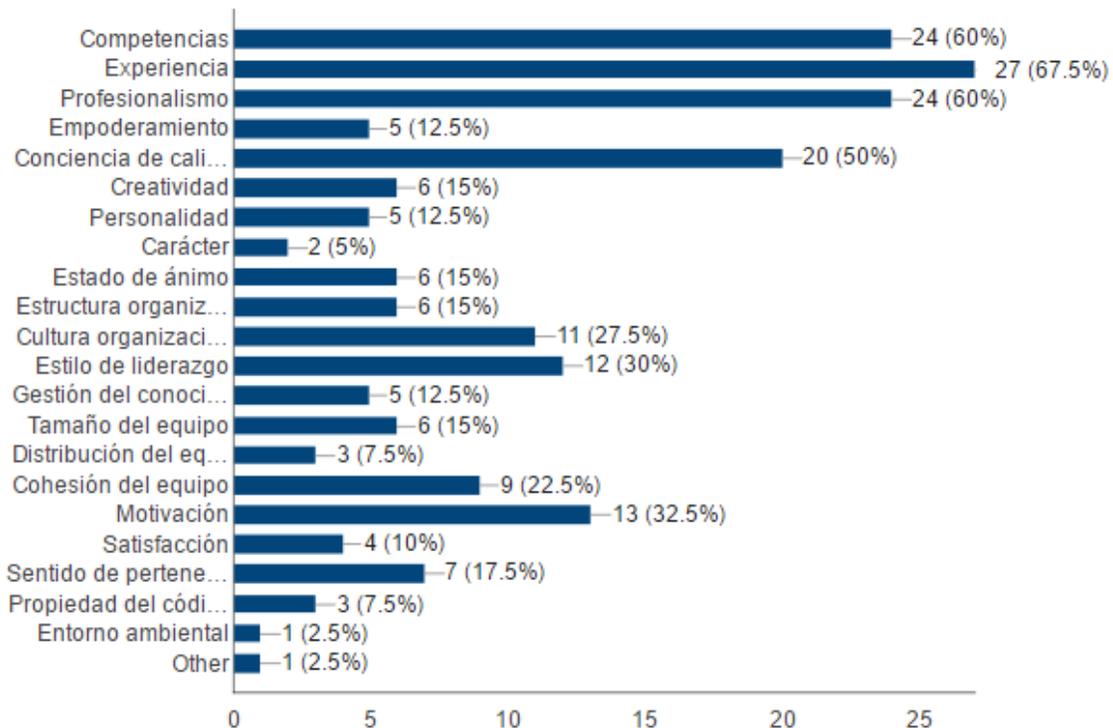
- 1) **Competencias:** contar con expertos en cada área.
- 2) **Carácter:** ser práctico y no perfeccionista.
- 3) **Estado de ánimo:** resiliencia o capacidad de sobreponerse a los períodos de dolor emocional y situaciones adversas, estado de salud.
- 4) **Estructura organizativa:** control, supervisión, administración efectiva del proyecto, área de aseguramiento de la calidad definida y estructurada, clara distribución por roles.
- 5) **Cultura organizacional:** el valor de la simplicidad y no complejidad, promover la discusión de las soluciones.
- 6) **Satisfacción:** ascensos y promociones, horarios de trabajo apropiados.
- 7) **Cohesión del equipo:** relaciones interpersonales entre el equipo.

4.2.1.6 VALORACIÓN DE LOS FACTORES HUMANOS QUE INCIDEN EN LA CALIDAD DEL SOFTWARE

Los 5 factores humanos que los participantes opinan en su mayoría que inciden en la calidad del software son:

- 1) La experiencia.
- 2) Las competencias.
- 3) El profesionalismo.
- 4) La conciencia de calidad.
- 5) Motivación.

FIGURA NO. 51 VALORACIÓN DE LOS FACTORES HUMANOS



Nota. Elaboración propia.

4.1 Análisis y discusión

En la fase cualitativa de la investigación se entrevistaron un total de 5 profesionales, y en la fase cuantitativa se encuestaron un total de 40 profesionales distribuidos en aproximadamente 20 empresas de El Salvador. De ellos el 70% fueron hombres, y el 30% mujeres; 40% tenían entre 21 y 30 años, 37.5% entre 31 y 40 años, y 22.5% más de 40 años; el 70% tenían un grado de licenciatura o ingeniería, el 25% grado de maestría, y el resto un grado técnico o de bachillerato. La mayoría de los participantes trabajaban en empresas privadas, grandes (de más de 100 empleados), y del sector terciario (comercio e industria). La mayoría ha trabajado como analista, programador o diseñador de software. Un 60% dijo tener 4 a 9 años de experiencia y el 30% más de 10 años de experiencia. El tamaño del equipo en el que han trabajado más frecuentemente es de 4 a 6 miembros, y en proyectos de una duración más frecuente de 7 a 12 meses.

El 62.5% dijo evaluar siempre la calidad del software que desarrollaba, el 35% algunas veces, y el resto nunca. Aspectos que indicaron evaluar con más frecuencia fue la conformidad con los requisitos y los errores o fallas. Con menos frecuencia la conformidad con los estándares de desarrollo y la eficiencia. Entre las técnicas de evaluación más comunes indicaron las pruebas y la retroalimentación del usuario. Con menos frecuencia la revisión de código y en un par de casos las métricas.

En cuanto al reconocimiento de los factores humanos, en la Tabla 23 se muestra cada factor humano y el porcentaje de participantes que considera que incide o no en la calidad del software.

TABLA NO. 23 RECONOCIMIENTO DE LOS FACTORES HUMANOS

Factores humanos	Incide	No incide
Competencias	100% (+)	0%
Experiencia	90% (+)	10%
Profesionalismo	97.5% (+)	2.5%
Empoderamiento	77.5% (+), 5% (-)	17.5%
Conciencia de calidad	97.5% (+)	2.5%
Creatividad	52.5% (+), 17.5% (-)	30%
Personalidad	55% (+)	45%
Carácter	72.5% (+)	27.5%
Estado de ánimo	77.5% (+), 22.5% (-)	0%
Estructura organizativa	87.5% (+ o -)	12.5%
Cultura organizacional	82.5% (+ o -)	17.5%
Estilo de liderazgo	97.5% (+ o -)	2.5%
Gestión del conocimiento	95% (+)	5%
Trabajo en equipo		
Tamaño del equipo	55% (-)	45%
Distribución del equipo	72.5% (+ o -)	27.5%
Cohesión del equipo	85% (+)	15%
Motivación	90% (+)	10%
Satisfacción	97.5% (+ o -)	2.5%
Sentido de pertenencia	80% (+)	20%
Propiedad del código	87.5% (+ o -)	12.5%
Entorno ambiental	87.5% (+ o -)	12.5%

Nota. Elaboración propia. El signo (+) indica que se considera que el factor incide positivamente en la calidad del software. El signo (-) que incide negativamente. Y los signos (+ o -) indica que se considera incide positiva o negativamente dependiendo de los aspectos que se describen en los siguientes párrafos.

La mayoría de los participantes coincide que las competencias que inciden positivamente en la calidad del software son: conocimiento técnico, conocimiento del negocio, pensamiento lógico y analítico, habilidad para aplicar conocimiento, habilidad para resolver problemas, habilidad investigativa, habilidad de comunicación, habilidad para tomar decisiones, atención al detalle y la motivación para crear o mejorar artefactos.

Entre los aspectos de la estructura organizativa que se considera inciden positivamente se indicaron con más frecuencia: flexibilidad de la organización, líneas de autoridad claramente definidas, responsabilidades claramente definidas, procedimientos claramente definidos, subdivisión del trabajo por roles, y la existencia de una unidad de aseguramiento de la calidad. La complejidad de la estructura organizativa es considerada por la mayoría que incide negativamente.

Los que opinan que la cultura organizacional incide en la calidad del software indican con más frecuencia que los aspectos que inciden positivamente son: cultura de planificación, cultura de compromiso con la calidad, clima de estabilidad, clima de respeto y confianza, cultura de seguir el proceso debido, y cultura de mejora continua. La cultura de búsqueda de culpables se considera incide negativamente por la totalidad de los participantes.

Los estilos de liderazgo que se considera inciden positivamente son: el liderazgo consultivo (el líder busca el consejo del equipo antes de tomar una decisión sobre lo que debe hacerse), el liderazgo participativo (el líder se ubica como miembro y discute la posible decisión con el equipo), y el liderazgo negociador (el líder utiliza incentivos para motivar al equipo a hacer ciertas cosas). Y los que inciden negativamente: el liderazgo directivo (el líder dice al equipo que hacer y cómo hacerlo), y el liderazgo delegativo (el líder deja que el equipo tome sus propias decisiones sobre lo que debe hacerse).

La mayoría de los participantes que respondieron que el tamaño del equipo incide en la calidad del software, opinan que el tamaño del equipo óptimo está entre los 4 y 5 miembros.

La mayoría de los participantes que respondieron que la distribución del equipo incide en la calidad del software, opinan que los equipos co-ubicados (los miembros trabajan en el mismo lugar) promueven la calidad del software. Mientras que los equipos globalizados o distribuidos (los miembros trabajan distribuidos geográficamente), y externalizados (equipos con responsabilidades diferentes que trabajan fuera de la organización) no la promueven.

Entre los aspectos relacionados con la satisfacción que se indica que inciden positivamente están: tener un salario apropiado, tener un plan de beneficios, la formación continua, la contratación directa, el contar con acceso a Internet, el contar con acceso al correo electrónico y los reconocimientos.

La mayoría de los participantes respondieron que la propiedad del código incide en la calidad del código, y que el modelo de propiedad que incide positivamente es el débil (el sistema es asignado a un desarrollador, pero se permite la modificación por otros bajo la supervisión del dueño). Mientras que los modelos fuerte (el sistema es propiedad de un solo desarrollador y solo él puede modificarlo) y colectivo (cualquiera puede hacer modificaciones) inciden negativamente en la calidad del software.

Entre los aspectos del entorno ambiental que se mencionan inciden positivamente en la calidad del software están: contar con equipo de computación apropiado, contar con amueblamiento apropiado, contar con espacios de trabajo privados y con espacios para descanso y socialización.

Los 5 factores humanos que los participantes opinan que inciden más en la calidad del software, en orden de importancia, son: la *experiencia*, las *competencias*, el *profesionalismo*, la *conciencia de calidad* y la *motivación*. Y los 5 factores humanos menos importantes en su relación con la calidad del software son: el *entorno ambiental*, el *carácter*, la *satisfacción*, la *propiedad del código* y la *distribución del equipo*.

La revisión de la literatura identificó 20 factores humanos. Para 18 de estos factores se encontró evidencia ya sea de fuentes primarias, secundarias o terciarias que indican que inciden o no inciden en la calidad del software.

Según estas fuentes, los factores humanos que inciden en la calidad del software son: las *Competencias*, el *Profesionalismo*, el *Empoderamiento*, la *Conciencia de calidad*, la *Creatividad*, el *Carácter*, el *Estado de ánimo*, la *Estructura organizativa*, la *Cultura organizacional*, el *Estilo de liderazgo*, la *Gestión del conocimiento*, el *Trabajo en Equipo (Tamaño y Cohesión solamente)*, la *Motivación*, la *Satisfacción*, el *Sentido de pertenencia*, la *Propiedad del código*, y el *Entorno ambiental*.

Los factores humanos que no inciden en la calidad del software son la *Experiencia* y la *Distribución del equipo* en el factor *Trabajo en Equipo*.

Se identificaron 2 factores humanos que no han sido estudiados en cuanto a su relación con la calidad del software: la *Gestión del cambio* y la *Personalidad* (Tabla 24).

El factor humano que más ha sido estudiado en su relación con la calidad del software es el de las *Competencias*. Seguido de la *Cultura organizacional*, el *Profesionalismo*, la *Creatividad*, el *Estado de ánimo*, la *Estructura organizativa*, el *Tamaño y Distribución del equipo*. Y los que menos han sido estudiados, con tan solo un estudio identificado, son: la *Experiencia*, el *Estilo de liderazgo*, la *Gestión del conocimiento*, la *Propiedad del código* y el *Entorno ambiental*. Factores humanos que no han sido estudiados en fuentes primarias son: el *Empoderamiento*, la *Conciencia de calidad*, la *Personalidad*, el *Carácter*, y la *Gestión del cambio*.

La investigación reveló que, a excepción de los factores humanos de la *Experiencia*, el *Tamaño y Distribución del equipo*, los factores humanos identificados tienen un buen grado de reconocimiento en el contexto salvadoreño por cuanto la experiencia de los profesionales salvadoreños confirma los hallazgos de la revisión de la literatura.

TABLA NO. 24 DISCUSIÓN DE LOS FACTORES HUMANOS

Factores humanos	¿Existen estudios que confirman la relación?	Incide según Revisión	Incide según profesionales
Competencias	Si (7)	Si	Si
Experiencia	Si (1)	No	Si
Profesionalismo	Si (3)	Si	Si
Empoderamiento	Si (0)	Si	Si
Conciencia de calidad	Si (0)	Si	Si
Creatividad	Si (3)	Si	Si
Personalidad	Si (9)	-	Si
Carácter	Si (0)	Si	Si
Estado de ánimo	Si (3)	Si	Si
Estructura organizativa	Si (3)	Si	Si
Cultura organizacional	Si (4)	Si	Si
Estilo de liderazgo	Si (1)	Si	Si
Gestión del cambio	No (0)	-	-
Gestión del conocimiento	Si (1)	Si	Si
Trabajo en equipo			
Tamaño del equipo	Si (3)	Si	Si/No
Distribución del equipo	Si (3)	No	Si/No
Cohesión del equipo	Si (2)	Si	Si
Motivación	Si (2)	Si	Si
Satisfacción	Si (2)	Si	Si
Sentido de pertenencia	Si (2)	Si	Si
Propiedad del código	Si (1)	Si	Si
Entorno ambiental	Si (1)	Si	Si

Nota. Elaboración propia. En la columna ¿Existen estudios que confirman la relación?, se detalla con un Sí o No si se identificaron en la revisión de la literatura fuentes primarias, secundarias o terciarias que establecen la relación entre el factor identificado y la calidad del software, y entre paréntesis se detalla el número de estudios primarios identificados.

Los profesionales salvadoreños consideran que la *Experiencia* incide en la calidad del software, contrario a lo que evidencian las investigaciones primarias estudiadas. Es de notar el hecho de que, consideran que la *Experiencia* es el factor que más incide en la calidad del software. Un hecho notorio es que, de forma unánime, en el caso de las *Competencias*, y casi unánime, en el caso del *Profesionalismo* los profesionales participantes indican que inciden en la calidad del software, y son los dos factores después de la *Experiencia* que se considera inciden más en la calidad del software. Lo que coincide con la idea de la inversión en la mejora de procesos, metodologías y técnicas es de

poca importancia cuando el empleado aplicándolas no tiene los conocimientos y habilidades apropiadas, o no tiene interés en utilizarlas [34, p. 74].

En cuanto al *Tamaño del equipo*, la opinión es dividida, poco más de la mitad de los profesionales considera que incide, mientras que el resto, que no incide. Esto contrasta con la revisión de la literatura que identifica que a mayor número de miembros en el equipo de trabajo la calidad del software es menor. Sin embargo, el número de miembros óptimo que los participantes identifican (4 a 5 miembros), coincide con lo que la literatura afirma. Y también en cuanto a la *Distribución del equipo*, la mayoría de los participantes considera que incide mientras que la revisión de la literatura indica que no es ese el caso. Esta discrepancia probablemente se debe al hecho de que ninguno de los estudios considera el modelo de distribución *externalizado*, el cual se está volviendo más popular en nuestros días, y para el cual amerita que se hagan más investigaciones.

En cuanto a la *Estructura organizativa* los profesionales salvadoreños consideran que el factor que más incide negativamente en la calidad del software es la complejidad de la estructura organizativa contrastando con la idea de Brooks de que a medida que las organizaciones se vuelven más complejas las comunicaciones se vuelven más difíciles y esto a su vez acarrea problemas que inciden en la calidad [115, p. 74].

Es interesante que no se indicara como *Estilo de liderazgo* que promueve la calidad, el estilo directivo, y se le considere como un estilo que incide negativamente. Esto contrasta con los estudios que indican que los dos estilos que promueven la calidad son el participativo y el directivo.

También en cuanto a la propiedad del código se reconoce la importancia de que el código sea asignado a un responsable para garantizar su calidad. Y que la conciencia de calidad, la motivación, la satisfacción del empleado y el sentirse parte de la organización son factores que promueven la calidad del software..

5 Conclusiones

En la investigación se ha buscado responder a la pregunta de qué factores humanos inciden en la calidad del software. Se han identificado, caracterizado y clasificado 20 factores humanos, y establecido cuáles de ellos han sido estudiados y cuáles no.

Sin embargo es evidente, a partir de los resultados, que la variedad y cantidad de factores humanos que inciden en la calidad del software es grande, y que, aún hay muchos por identificar. Además, la incidencia de algunos no ha sido establecida del todo. El tema puede y debe seguir investigándose. Por ejemplo, la incidencia de factores como la *Experiencia*, el *Estilo de liderazgo*, la *Gestión del conocimiento*, la *Propiedad del código* y el *Entorno ambiental*, por no estar establecida de forma concluyente amerita más investigación empírica. También, dado que se reconoce la importancia de factores como el *Empoderamiento*, la *Conciencia de calidad*, la *Personalidad*, el *Carácter*, y la *Gestión del cambio*, es importante que se confirme su incidencia con estudios empíricos.

También, la investigación ha dado respuesta a la pregunta de qué buenas prácticas relacionadas con los factores humanos identificados posibilitan el desarrollo de software de calidad, y a la pregunta de qué indicadores pueden ser utilizados para evaluar los equipos y organizaciones como parte de los procesos del control y aseguramiento de la calidad.

Los resultados indican además, que existe en el contexto salvadoreño un buen grado de conciencia de los factores humanos y de cómo estos inciden en la calidad del software. Pero, consideramos que este reconocimiento debe ampliarse a más profesionales utilizando de preferencia métodos de recolección de datos cualitativos como la observación o las entrevistas.

Aunque la investigación ha establecido la importancia de los factores humanos en la calidad del software, y que factores humanos se reconoce y existe evidencia empírica de que inciden en la calidad del software, es importante tener presente que además de las personas, los procesos son también un elemento clave en el desarrollo de software. Procesos de desarrollo, de gestión de proyecto de desarrollo de software, de control y de aseguramiento de la calidad son, como hemos visto, necesarios para alcanzar un producto de software de calidad. Las organizaciones deben contar no solo con el recurso humano idóneo, que tenga los conocimientos, las habilidades apropiadas y el interés por utilizarlas, sino también con procesos establecidos que garanticen su aplicación correcta y contribuyan al desarrollo de un producto de software de calidad.

Finalmente, la investigación reveló además la necesidad de otros posibles estudios relacionados con la calidad del software: estudios para documentar y evaluar la situación actual de la calidad del software en relación a la idea que se tiene de ella, los aspectos que se evalúan para determinar si un producto de software es de calidad o no, y las profundizar en las técnicas y métodos que se utilizan para evaluarla.

6 Bibliografía

- [1] C. Camisón, S. Cruz y T. González, *Gestión de la calidad: conceptos, enfoques, modelos y sistemas*, 1a. ed., Madrid: Pearson Educación S.A., 2006.
- [2] M. Piattini, F. García y I. Caballero, *Calidad de Sistemas Informáticos*, 1a. ed., México D.F.: Alfaomega Grupo Editor S.A. de C.V., 2007.
- [3] J. Evans y W. Lindsay, *Administración y control de la calidad*, 7a. ed., México D.F.: Cengage Learning, 2008.
- [4] J. Juran, *Quality Handbook*, 5ta. ed., Nueva York: Mc Graw Hill, 1999.
- [5] A. Gillies, *Software Quality: Theory and management*, 1a. ed., Londres: Chapman & Hall, 1992.
- [6] D. Besterfield, *Control de Calidad*, 8a. ed., México D.F.: Pearson Education, 2009.
- [7] ISO, «ISO 9001 Sistemas de Gestión de Calidad - Requisitos,» 2008. [En línea]. Available: <http://farmacia.unmsm.edu.pe/noticias/2012/documentos/ISO-9001.pdf>. [Último acceso: 12 junio 2015].
- [8] D. Garvin, «Competing on the eight dimensions of quality,» noviembre 1987. [En línea]. Available: <https://hbr.org/1987/11/competing-on-the-eight-dimensions-of-quality>. [Último acceso: 15 febrero 2016].
- [9] D. Garvin, «What does "product quality" really mean?,» octubre 1984. [En línea]. Available: <http://sloanreview.mit.edu/article/what-does-product-quality-really-mean/>. [Último acceso: 15 febrero 2016].
- [10] S. Sánchez, M. Sicilia y D. Rodríguez, *Ingeniería del Software. Un enfoque desde la guía SWEBOK*, 1a. ed., Madrid: Alfaomega Grupo Editor S.A. de C.V., 2012.
- [11] R. Pressman, *Ingeniería de Software*, 7a. ed., D.F.: McGraw-Hill Interamericana Editores S.A. de C.V., 2010.
- [12] I. Sommerville, *Ingeniería de Software*, 9a. ed., México D.F.: Pearson Education de México S.A. de C.V., 2011.
- [13] T. DeMarco y T. Lister, *Peopleware: Productive projects and teams*, 3a. ed., Reading, Massachusetts: Addison-Wesley Professional, 2013.
- [14] F. P. Brooks, *The Mythical Man-Month, Anniversary ed.*, Reading, Massachusetts: Addison Wesley Logman Inc., 1995.
- [15] D. Chappell, «The three aspects of software quality: functional, structural and process,» s.f.. [En línea]. Available: http://www.davidchappell.com/writing/white_papers/The_Three_Aspects_of_Software_Quality_v1.0-Chappell.pdf. [Último acceso: 25 abril 2016].
- [16] BSA, «Software Industry Facts and Figure,» 2010. [En línea]. Available: http://www.bsa.org/country/public%20policy/~~/media/files/policy/security/general/sw_factsfigur es.ashx. [Último acceso: 19 noviembre 2015].

- [17] SIIA, «The U.S. Software Industry: An Engine for Economic Growth and Employment,» 2014. [En línea]. Available: <https://www.sii.net/Admin/FileManagement.aspx/LinkClick.aspx?fileticket=yLPW0SrBfk4%3D&portalid=0>. [Último acceso: 20 junio 2016].
- [18] CEPAL, «Estado de la Banda Ancha en América Latina y el Caribe,» 2012. [En línea]. Available: <http://archivo.cepal.org/pdfs/2012/S2012095.pdf>. [Último acceso: 19 noviembre 2015].
- [19] El Diario de Hoy, «Solo el 20 % en el país tiene acceso a Internet,» 16 mayo 2013. [En línea]. Available: <http://www.elsalvador.com/articulo/entretenimiento/solo-pais-tiene-acceso-internet-32800>. [Último acceso: 19 noviembre 2015].
- [20] Universia, «II Estudio de Redes Sociales en América Central,» 18 marzo 2013. [En línea]. Available: <http://noticias.universia.com.sv/en-portada/noticia/2013/03/18/1011577/salvador-tiene-mas-1-5-millones-usuarios-facebook.html>. [Último acceso: 19 noviembre 2015].
- [21] IEEE, «IEEE Standard Glossary of Software Engineering Terminology STD 610.12-1990,» 1990. [En línea]. Available: http://dis.unal.edu.co/~icasta/GGP/_Ver_2013_1/Documentos/Normas/610-12-1990.pdf. [Último acceso: 18 junio 2015].
- [22] W. S. Humphrey, «The Software Quality Profile,» 2009. [En línea]. Available: <http://www.sei.cmu.edu/library/assets/softwarequalityprofile.pdf>. [Último acceso: 18 junio 2015].
- [23] IEEE, SWEBOOK V3.0 Guide to the Software Engineering Book of Knowledge, 1a. ed., Nueva Jersey: IEEE Computer Society, 2014.
- [24] G. Pantaleo, Calidad en el desarrollo de software, 1a. ed., Argentina: Alfaomega Grupo Editor Argentino, 2011.
- [25] S. McConnell, Code Complete, 2da. ed., Redmond, Washington: Microsoft Press, 2004.
- [26] G. O'Regan, A Practical Approach to Software Quality, New York: Springer-Science+Business Media, LLC, 2002.
- [27] N. Dershowitz, «Software Horror Stories,» s.f.. [En línea]. Available: <http://www.cs.tau.ac.il/~nachumd/horror.html>. [Último acceso: 21 noviembre 2015].
- [28] S. Aguilar, «20 desastres famosos relacionados con el software,» 23 noviembre 2008. [En línea]. Available: <http://www.variabilenotfound.com/2008/11/20-desastres-famosos-relacionados-con.html>. [Último acceso: 21 noviembre 2015].
- [29] CHFG, «What is human factors?,» s.f.. [En línea]. Available: <http://chfg.org/what-is-human-factors>. [Último acceso: 19 noviembre 2015].
- [30] S. Robbins y T. Judge, Comportamiento organizacional, 15a. ed., México D.F.: Pearson Education de México S.A. de C.V., 2013.
- [31] F. J. Hemstra, «Soft Factors affecting Software Quality,» de *Fundamental Concepts for the Software Quality Engineer, Volumen 2*, 1a. ed., Milwaukee, Wisconsin: American Society for Quality (ASQ), 2001, pp. 47-64.
- [32] FAA, «Aviation Maintenance Technician Handbook (AMT),» 2008. [En línea]. Available: https://www.faa.gov/regulations_policies/handbooks_manuals/aircraft/amt_handbook/. [Último

acceso: 20 abril 2016].

- [33] HSE, «Introduction to human factors (Human Factors Toolkit),» 2005. [En línea]. Available: <http://www.hse.gov.uk/humanfactors/topics/toolkitintro.pdf>. [Último acceso: 20 abril 2016].
- [34] P. Merrill, «Getting the best out of people: ISO 10018 aids ISO 9001 implementation,» 2012. [En línea]. Available: <http://www.bsj.org.jm/Portals/0/ncbj/ISO%2010018%20The%20best%20of%20people.pdf>. [Último acceso: 12 junio 2015].
- [35] B. Curtis, «Top Five Causes of Poor Software Quality,» 1 julio 2009. [En línea]. Available: <http://www.datamation.com/entdev/article.php/3827841/Top-Five-Causes-of-Poor-Software-Quality.htm>. [Último acceso: 20 noviembre 2015].
- [36] M. Lavallée y P. Robillard, «Why Good Developers Write Bad Code: An Observational Case Study of the Impacts of Organizational Factors on Software Quality,» 2015. [En línea]. Available: http://www.upedu.org/papers/ICSE2015_OrganizationalFactors/LavalleeRobillard_ICSE2015_WhyGoodDevelopersWriteBadCode.pdf. [Último acceso: 25 junio 2015].
- [37] N. Nagappan, B. Murphy y V. Basili, «The influence of organizational structure on software quality: an empirical case study,» 2008. [En línea]. Available: <http://research.microsoft.com/pubs/70535/tr-2008-11.pdf>. [Último acceso: 12 junio 2015].
- [38] M. Prifling, «Exploring leadership styles in software development projects,» 2010. [En línea]. Available: <http://www.pacis-net.org/file/2010/S03-04.pdf>. [Último acceso: 26 junio 2015].
- [39] V. R. Basili y G. Caldiera, «Improve Software Quality by Reusing Knowledge and Experience,» 1995. [En línea]. Available: <https://www.cs.umd.edu/~basili/publications/journals/J55.pdf>. [Último acceso: 23 junio 2015].
- [40] A. Meneely y L. Williams, «Secure Open Source Collaboration: An Empirical Study of Linus' Law,» 2009. [En línea]. Available: <http://www4.ncsu.edu/~apmeneel/ccs221-meneely.pdf>. [Último acceso: 23 junio 2015].
- [41] D. Mahajan, «A Study of Correlation of Software Development Project's Team Size with its Quality Variances,» 2012. [En línea]. Available: http://www.ijbui.com/papers/vol2issue1/ijbi/Jun_13_IJBI_003.pdf. [Último acceso: 28 junio 2015].
- [42] C. Bird, N. Nagappan, B. Murphy, H. Gall y P. Devanbu, «Does Distributed Development Affect Software Quality? An Empirical Case Study of Windows Vista,» 2009. [En línea]. Available: <http://macbeth.cs.ucdavis.edu/distributed.pdf>. [Último acceso: 28 junio 2015].
- [43] J. W. Creswell, *Research Design: Qualitative, Quantitative and Mixed Methods Approaches*, 4a. ed., Thousand Lakes, California: Sage Publications Inc., 2014.
- [44] R. Hernández Sampieri, C. Fernández Collado y P. Baptista Lucio, *Metodología de la Investigación*, 6a. ed., D.F.: McGraw-Hill Interamericana, S.A. de C.V., 2014.
- [45] M. Mason, «Sample Size and Saturation in PhD Studies Using Qualitative Interviews,» 2010. [En línea]. Available: <http://www.qualitative-research.net/index.php/fqs/article/view/1428/3027>. [Último acceso: 7 diciembre 2015].
- [46] W. Booth, G. Colomb y J. Williams, *The Craft of Research*, 3a. ed., Chicago: The University of

Chicago Press, 2008.

- [47] Merriam-Webster, «Knowledge,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/knowledge>. [Último acceso: 20 junio 2015].
- [48] Merriam-Webster, «Skill,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/skill>. [Último acceso: 20 junio 2015].
- [49] H. Sackman, W. J. Erikson y E. E. Grant, «Exploratory experimental studies comparing online and offline programming performance,» *Communications of the ACM*, vol. 11, nº 1, pp. 3-11, enero 1968.
- [50] B. Curtis, «Substantiating Programmer Variability,» *Proceedings of the IEEE*, vol. 69, nº 7, p. 846, julio 1981.
- [51] H. D. Mills, *Software Productivity*, Nueva York, Nueva York: Dorset House Publishing Company, 1988.
- [52] T. DeMarco y T. Lister, «Programmer performance and the effects of the workplace,» *ICSE '85 Proceedings of the 8th international conference on Software Engineering*, pp. 268-272, 1985.
- [53] B. Curtis, E. M. Soloway, R. E. Brooks, J. B. Black, K. Ehrlich y H. R. Ramsey, «Software psychology: The need for an interdisciplinary program,» *Proceedings of the IEEE*, vol. 74, nº 8, pp. 1092-1106, agosto 1986.
- [54] D. N. Card, «A software technology evaluation program,» *Information and Software Technology*, vol. 29, nº 6, pp. 291-300, julio 1987.
- [55] B. W. Boehm y P. Papaccio, «Understanding and Controlling Software Costs,» *IEEE Transactions on Software Engineering*, vol. 14, nº 10, pp. 1462-1477, octubre 1988.
- [56] J. D. Vallet y F. E. McGarry, «A summary of software measurement experiences in the software engineering laboratory,» *Journal of Systems and Software*, vol. 9, nº 2, pp. 137-148, febrero 1989.
- [57] B. Boehm, C. Abts, A. W. Brown, S. Chulani, B. K. Clark, E. Horowitz, R. Madachy, D. J. Reifer y B. Steece, *Software Cost Estimation with Cocomo II*, 1a. ed., Upper Saddle River, Nueva Jersey: Prentice Hall, 2000.
- [58] A. Hunt y D. Thomas, *The Pragmatic Programmer*, 1a. ed., Reading, Massachusetts: Addison-Wesley, 2000.
- [59] SEI, «A Software Engineering Body of Knowledge Version 1.0,» 1999. [En línea]. Available: https://resources.sei.cmu.edu/asset_files/TechnicalReport/1999_005_001_16733.pdf. [Último acceso: 20 junio 2015].
- [60] ACM, «Computing Degrees and Careers,» 2015a. [En línea]. Available: http://computingcareers.acm.org/?page_id=6. [Último acceso: 20 junio 2015].
- [61] ACM-IEEE, «Curricula Recommendations,» 2015. [En línea]. Available: <https://www.acm.org/education/curricula-recommendations>. [Último acceso: 20 junio 2015].
- [62] ACM, «Skills you'll learn if you study computing,» 2015b. [En línea]. Available: http://computingcareers.acm.org/?page_id=15. [Último acceso: 20 junio 2015].
- [63] C. Anderson y J. F. Gantz, «Skills requirements for tomorrow's best jobs,» 2013. [En línea].

- Available: http://news.microsoft.com/download/presskits/education/docs/idc_101513.pdf. [Último acceso: 23 junio 2015].
- [64] A. Hunt y D. Thomas, *The Pragmatic Programmer*, 1a. ed., Reading, Massachusetts: Addison-Wesley, 2000.
- [65] Merriam-Webster, «Experience,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/experience>. [Último acceso: 20 junio 2015].
- [66] Merriam-Webster, «Professionalism,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/professionalism>. [Último acceso: 20 junio 2015].
- [67] D. Miller, «Software Quality Requires Professionalism and Fortitude,» 2002. [En línea]. Available: http://rube.asq.org/pub/sqp/past/vol4_issue4/sqp4i4miller.pdf. [Último acceso: 27 junio 2015].
- [68] G. Ford y N. E. Gibbs, «A mature profession of software engineering,» 1996. [En línea]. Available: <http://www.sei.cmu.edu/reports/96tr004.pdf>. [Último acceso: 21 junio 2015].
- [69] ASIA, «Código de Ética,» 1998. [En línea]. Available: <http://asiasv.org/codigo-de-etica/>. [Último acceso: 21 junio 2015].
- [70] ACM-IEEE, «Software Engineering of Ethics and Profesional Practices,» 1999. [En línea]. Available: <https://www.acm.org/about/se-code-s>. [Último acceso: 20 junio 2015].
- [71] Merriam-Webster, «Commitment,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/commitment>. [Último acceso: 23 junio 2015].
- [72] K. V. Siakas y E. Georgiadou, «The Role of Commitment for successful Software Process Improvement and Software Quality Management,» 2002. [En línea]. Available: http://www.researchgate.net/profile/Kerstin_Siakas/publication/228560145_The_Role_of_Commitment_for_successful_Software_Process_Improvement_and_Software_Quality_Management/links/0fcfd50cb194e29fd2000000.pdf?disableCoverPage=true. [Último acceso: 28 junio 2015].
- [73] P. Abrahamsson, «Rethinking the Concept of Commitment in Software Process Improvement,» 2001. [En línea]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.473.9865&rep=rep1&type=pdf>. [Último acceso: 28 junio 2015].
- [74] ASQ, «Employee Empowerment,» 2015a. [En línea]. Available: <http://asq.org/learn-about-quality/employee-involvement/overview/overview.html>. [Último acceso: 12 junio 2015].
- [75] R. Nandyal, *Making Sense of Software Quality Assurance*, 1a. ed., Nueva Delhi: McGraw Hill, 2007.
- [76] Merriam-Webster, «Awareness,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/awareness>. [Último acceso: 20 junio 2015].
- [77] Merriam-Webster, «Creativity,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/creativity>. [Último acceso: 27 junio 2015].
- [78] M. G. Xin Tong, «Towards Hypotheses on Creativity in Software Development,» 2004. [En línea]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.107.5769&rep=rep1&type=pdf>.

[Último acceso: 27 junio 2015].

- [79] A. B. Godfrey, «Creativity, Innovation and Quality,» 1996. [En línea]. Available: <http://leonardopublic.innovation.si/4.Techniques%20of%20idea%20creation/Creativity,%20Innovation%20and%20Quality.pdf>. [Último acceso: 28 junio 2015].
- [80] APA, «Personality,» 2016. [En línea]. Available: <http://www.apa.org/topics/personality/>. [Último acceso: 15 julio 2016].
- [81] Neris Analytic Limited, «Nuestra teoría,» 2016. [En línea]. Available: <https://www.16personalities.com/es/nuestra-teoria>. [Último acceso: 15 julio 2016].
- [82] Neris Analytics Limited, «Tipos de personalidad,» 2016. [En línea]. Available: <https://www.16personalities.com/es/descripcion-de-los-tipos>. [Último acceso: 15 julio 2016].
- [83] L. F. Capretz, «Personality Types in Software Engineering,» febrero 2003. [En línea]. Available: http://eng.uwo.ca/electrical/faculty/mcisaac_k/docs/mbti-IJHCS-v2.pdf. [Último acceso: 25 julio 2016].
- [84] A. Raza, L. Capretz y Z. Mustafa, «Personality Profiles of Software Engineers and Their Software Quality Preferences,» s.f.. [En línea]. Available: <https://arxiv.org/ftp/arxiv/papers/1511/1511.04411.pdf>. [Último acceso: 15 julio 2016].
- [85] S. T. Acuña, M. N. Gómez, J. E. Hannay, N. Juristo y D. Pfhal, «Are team personality and climate related to satisfaction and software quality? Aggregating results from a twice replicated experiment,» *Information and Software Technology*, vol. 57, pp. 141-156, 2015.
- [86] Merriam-Webster, «Character,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/character>. [Último acceso: 23 junio 2015].
- [87] D. Graziotin, Z. Wang y P. Abrahamsson, «Happy software developers solve problems better: psychological measurements in empirical software engineering,» 2014. [En línea]. Available: <https://peerj.com/articles/289.pdf>. [Último acceso: 30 junio 2015].
- [88] Business Dictionary, «Organizational structure,» [En línea]. Available: <http://www.businessdictionary.com/definition/organizational-structure.html>. [Último acceso: 28 junio 2015].
- [89] A. April y C. Laporte, «An Overview of Software Quality Concepts and Management Issues,» 2005. [En línea]. Available: http://www.etsmtl.ca/professeurs/claporte/documents/publications/duggan_chapter_sqa.pdf. [Último acceso: 18 junio 2015].
- [90] K. Wiegers, «Creating a Software Engineering Culture,» *Software Development Magazine*, 1994.
- [91] Business Dictionary, «Leadership,» [En línea]. Available: <http://www.businessdictionary.com/definition/leadership.html>. [Último acceso: 27 junio 2015].
- [92] WSJ, «Five Leadership Styles,» 2016. [En línea]. Available: <http://guides.wsj.com/management/developing-a-leadership-style/how-to-develop-a-leadership-style/>. [Último acceso: 15 junio 2016].
- [93] P. Berander, L. O. Damm, J. Eriksson, T. Gorscheck, K. Henningsson, P. Jonsson, S. Kagstrom, D. Milicic, F. Martensson, K. Ronkkon y P. Tomaszewski, «Software quality attributes and trade-offs,» 2005. [En línea]. Available:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.5016&rep=rep1&type=pdf>.
[Último acceso: 26 junio 2015].

- [94] ASQ, «What is leadership?,» 2015b. [En línea]. Available: <http://asq.org/learn-about-quality/leadership/overview/overview.html>. [Último acceso: 21 junio 2015].
- [95] P. D. Bachiochi, S. G. Rogelberg, M. S. O'Connor y A. E. Elder, «The qualities of an effective team leader,» 2000. [En línea]. Available: <http://orgscience.uncc.edu/sites/orgscience.uncc.edu/files/The%20Qualities%20of%20an%20Effective%20Team%20Leader.pdf>. [Último acceso: 26 junio 2015].
- [96] Business Dictionary, «Change Management,» [En línea]. Available: <http://www.businessdictionary.com/definition/change-management.html>. [Último acceso: 27 junio 2015].
- [97] ASQ, «What is change management?,» 2015c. [En línea]. Available: <http://asq.org/learn-about-quality/change-management/overview/overview.html>. [Último acceso: 21 junio 2015].
- [98] Forrester Consulting, «The Challenges Of Software Change Management In Today's Siloed IT Organizations,» 2006. [En línea]. Available: <http://www.serena.com/docs/repository/solutions/software-change-mana.pdf>. [Último acceso: 27 junio 2015].
- [99] Business Dictionary, «Knowledge Management,» [En línea]. Available: <http://www.businessdictionary.com/definition/knowledge-management.html>. [Último acceso: 23 junio 2015].
- [100] C. Bird, N. Nagappan, B. Murphy, H. Gall y P. Devanbu, «Don't Touch My Code! Examining the Effects of Ownership on Software Quality,» 2011. [En línea]. Available: <http://www.msri-waypoint.net/pubs/150498/bird2011dtm.pdf>. [Último acceso: 23 Junio 2015].
- [101] E. Weimar, A. Nugroho, J. Visser y A. Plaats, «Towards High Performance Software Teamwork,» 2013. [En línea]. Available: <https://askeplaats.files.wordpress.com/2013/01/final-towards-high-performance-software-teamwork-paper-for-the-2013-ease-conference.pdf>. [Último acceso: 12 junio 2015].
- [102] Merriam-Webster, «Communication,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/communication>. [Último acceso: 28 junio 2015].
- [103] V. Vinod, «Impact of team skills on software quality: A comparative study of software development teams in India and USA,» 2010. [En línea]. Available: http://www.drmgrdu.ac.in/Thesis_doc/Documents/Viji%20Vinod%20Ph.d%20Thesis/Viji%20VinodThesis.pdf. [Último acceso: 12 junio 2015].
- [104] I. H. Farias, L. Duarte, J. P. N. Oliveira, A. R. N. Dantas y J. F. Barbosa, «Motivational Factors for Distributed Software Development Team,» 2012. [En línea]. Available: <http://www.lbd.dcc.ufmg.br/colecoes/wdds/2012/003.pdf>. [Último acceso: 23 junio 2015].
- [105] D. Spinellis, «Global Software Development in the FreeBSD Project,» 2006. [En línea]. Available: <http://dl.acm.org/authorize?N95982>. [Último acceso: 28 junio 2015].
- [106] Merriam-Webster, «Motivation,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/motivation>. [Último acceso: 22 junio 2015].
- [107] A. Sajid, H. Nasir, A. Shehzad y Z. Mehtab, «Impact of Culture Based on Human and Soft

- Issues on Software Quality in Pakistan,» 2008. [En línea]. Available: <http://www.ep.liu.se/ecp/033/065/ecp0803365.pdf>. [Último acceso: 21 junio 2015].
- [108] Merriam-Webster, «Satisfaction,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/satisfaction>. [Último acceso: 28 junio 2015].
- [109] D. Graziotin, Z. Wang y P. Abrahamsson, «Happy software developers solve problems better: psychological measurements in empirical software engineering,» 2014. [En línea]. Available: <https://peerj.com/articles/289.pdf>. [Último acceso: 30 junio 2015].
- [110] M. Samuelsson, Personality Types and Attributes in Software Engineering, Trollhattan: University of Trollhattan (Department of Informatics, Mathematics and Computer Science), 2005.
- [111] K. Hall, «Create A Sense of Belonging,» 2014. [En línea]. Available: <https://www.psychologytoday.com/blog/pieces-mind/201403/create-sense-belonging>. [Último acceso: 28 junio 2015].
- [112] A. Sajid, H. Nasir, A. Shehzad y Z. Mehtab, «Impact of Culture Based on Human and Soft Issues on Software Quality in Pakistan,» 2008. [En línea]. Available: <http://www.ep.liu.se/ecp/033/065/ecp0803365.pdf>. [Último acceso: 21 junio 2015].
- [113] Merriam-Webster, «Ownership,» [En línea]. Available: <http://www.merriam-webster.com/dictionary/ownership>. [Último acceso: 23 junio 2015].
- [114] C. Bird, N. Nagappan, B. Murphy, H. Gall y P. Devanbu, «Don't Touch My Code! Examining the Effects of Ownership on Software Quality,» 2011. [En línea]. Available: <http://www.msri-waypoint.net/pubs/150498/bird2011dtm.pdf>. [Último acceso: 23 Junio 2015].
- [115] F. P. Brooks, The Mythical Man-Month, Essays on Software Engineering, Anniversary ed., Reading, Massachusetts: Addison-Wesley, 1995.

7 Glosario

Actitud. Enunciados o juicios que evalúan objetos, individuos o eventos [30, p. 71].

Actividad. Unidad mínima de trabajo, la cual tiene una duración definida, está relacionada lógicamente con otras actividades, consume recursos y tiene generalmente un costo asociado [10, p. 46].

Adaptabilidad. La medida en que el sistema puede ser usado, sin modificación en aplicaciones o ambientes diferentes de aquel para el cual fue diseñado [25, p. 465].

Aptitud. Capacidad de un individuo para realizar las distintas tareas de un puesto de trabajo [30, p. 53].

Aptitud intelectual. Ver Capacidad intelectual.

Aptitud física. Capacidad para realizar tareas que demandan vigor, destreza, fuerza y otras características similares [30, p. 55].

Artefacto. Algo tangible creado con un propósito práctico [10, p. 20].

Atributo. Característica.

Atributo externo. Los atributos de calidad externos del software son aquellas propiedades del software fácilmente observables por los usuarios [11, p. 94].

Atributo externo de calidad del software. Ver Atributo externo.

Atributo interno. Los atributos de calidad internos del software son de importancia para los ingenieros de software y desarrolladores. Conducen a un software de alta calidad desde el punto de vista técnico [11, p. 94].

Atributo interno de calidad del software. Ver Atributo interno.

Calidad. Las características de un producto o servicio que inciden en su capacidad para satisfacer las necesidades explícitas o implícitas, o un producto o servicio que está libre de deficiencias [6, p. 2].

Calidad de software. a) Las características deseadas de los productos de software, b) El grado en que un producto de software particular posee dichas características, c) Los procesos, herramientas y técnicas usados para alcanzar esas características [23, pp. 10-1].

Calidad del producto de software. La calidad del producto apunta a sus características y viene definida como el grado de observancia de un conjunto de criterios o atributos de calidad preestablecidos y medibles [10, p. 384].

Calidad del proceso de software. La calidad del proceso identifica la calidad con la calidad del proceso de producción. Todo producto fabricado de acuerdo a estándares regulados de calidad podrá ser considerado un producto de calidad [10, p. 383].

Calidad objetiva. Deriva de la comparación entre un estándar y n desempeño referidos a características de calidad medibles cuantitativamente. Describe ya sea la excelencia o la superioridad técnica de los atributos del producto o del proceso, siendo independiente de la persona que realiza la medición o adquiere el producto [1, p. 149].

Calidad subjetiva. Se basa en la percepción y en los juicios de valor de las personas, y es medible cualitativamente estudiando la satisfacción del cliente [1, p. 149].

Capacidad intelectual. Habilidad para realizar actividades mentales como pensamiento, razonamiento y solución de problemas [30, p. 53].

- Ciclo de vida.** El ciclo de vida de un proceso o producto es la evolución del mismo desde el momento de su concepción hasta el momento en que deja de usarse, y puede describirse en función de las actividades que se realizan dentro de él [10, p. 22].
- Ciclo de vida de desarrollo de software.** Período de tiempo que comienza cuando se toma la decisión de desarrollar un producto de software y que concluye cuando se entrega [10, p. 34].
- Comprensible.** Facilidad para entender el propósito y estructura del software [25, p. 464]
- Compromiso organizacional.** Grado en que un empleado se identifica con una organización particular y con sus metas, y desea seguir siendo miembro de ésta [30, p. 75].
- Corrección.** El grado en que el sistema está libre de errores en su especificación, diseño e implementación, si fue construido correctamente. Que las funcionalidades solicitadas en su especificación se encuentren disponibles [25, p. 465].
- Eficiencia.** Mínimo uso de recursos de sistema durante su funcionamiento, incluyendo memoria y tiempo de ejecución [25, p. 465].
- Especificación.** Es una descripción detallada y precisa de algo existente (o que existirá) o de una cierta situación, presente o futura [10, p. 21].
- Exactitud.** El grado en el cual el sistema, tal como fue construido, está libre de errores, especialmente con respecto a resultados cuantitativos, que tan bien el sistema hace el trabajo para el cual fue diseñado [25, p. 465].
- Fiabilidad.** La habilidad del sistema de ejecutar su funcionalidad bajo condiciones establecidas cuando sea requerido, tener un amplio tiempo medio entre fallas, ausencia de defectos [25, p. 465].
- Factor humano.** Todos aquellos elementos que pueden influir en las personas y su comportamiento [29].
- Factor humano en el trabajo.** Son los factores del medio ambiente, de la organización y de las interacciones con otros, y las características individuales que influyen en el comportamiento en el trabajo [29].
- Flexibilidad.** La medida con la que se puede modificar un sistema para usos diferentes para los cuales fue diseñado. Capacidad para introducir cambios en función de las necesidades del negocio [25, p. 464].
- Involucramiento en el trabajo.** Grado en que un individuo se identifica con su puesto de trabajo, participa activamente en él y considera su desempeño como algo importante que lo beneficia [30, p. 75].
- ISO 9000.** Proporciona una serie de prácticas excelentes para iniciar un sistema de calidad, y es un punto de partida para las empresas que no tienen un programa formal de aseguramiento de la calidad [3, p. 137].
- Integridad.** El grado en que el sistema previene el acceso no autorizado o inapropiado a sus programas y sus datos. La idea de integridad incluye el restringir el acceso no autorizado así como asegurar que la información sea usada apropiadamente [25, p. 465].
- Legibilidad.** La facilidad con la que puede ser leído y entendido el código fuente de un sistema [25, p. 464].
- Mantenibilidad:** Facilidad con la cual un software puede ser modificado para cambiar o añadir funcionalidades, mejorar el desempeño o corregir defectos. Disposición para ser modificado para ser corregido, adaptado o ampliado [25, p. 464].

- Método.** Especificación de una secuencia de acciones orientadas a un propósito determinado. Los métodos determinan el orden y la forma de llevar a cabo las actividades [10, p. 21].
- Metodología.** Conjunto de métodos coherentes y relacionados por unos principios comunes [10, p. 21].
- Personalidad.** Características perdurables que describen el comportamiento de un individuo [30, p. 133].
- Poder ser probado.** El grado en que se pueden hacer pruebas unitarias o de sistema, capacidad de validar los requisitos establecidos para el software [25, p. 464].
- Portabilidad.** Facilidad para usar el software en un nuevo entorno (sistemas operativos, bases de datos, etc.) [25, p. 464].
- Proceso.** Conjunto de recursos y actividades interrelacionados que posibilitan la transformación de elementos de entrada en resultados. Los elementos de entrada para un proceso suelen ser los resultados de otros procesos. Los procesos pueden subdividirse en subprocesos [1, p. 148]. Secuencia de actividades que tiene como objetivo lograr un resultado [3, p. 21]. Una serie de pasos que incluyen actividades, restricciones y recursos que resultan en un producto determinado con ciertas características [10, p. 33].
- Proceso de desarrollo de software.** Ver Proceso de software.
- Proceso de software.** Conjunto coherente de políticas, estructuras organizativas, tecnologías, procedimientos, actividades y artefactos que se necesitan para concebir, desarrollar, implantar y mantener un producto de software [10, p. 22].
- Producto.** El resultado de un proceso. Pueden ser tangibles (bienes) o intangibles (servicios) [1, p. 148]. Todos los bienes y servicios que una empresa ofrece al cliente [1, p. 164].
- Proyecto.** Es un esfuerzo que se lleva a cabo una sola vez, que tiene objetivos bien definidos y que se produce dentro de un plazo determinado [10, p. 33].
- Requisito.** Ver Especificación.
- Reusabilidad.** La medida con la que y la facilidad con que pueden usarse partes de un sistema en otro sistema [25, p. 464].
- Robustez.** El grado en que el sistema continúa funcionando ante la presencia de entradas incorrectas o condiciones adversas [25, p. 465].
- Software.** El software es: a) instrucciones (programas de cómputo) que cuando se ejecutan proporcionan las características, función y desempeño buscados; b) estructuras de datos que permiten que los programas manipulen la información, y c) información descriptiva tanto en papel como en formas virtuales que describe la operación y uso de los programas" [11, p. 3].
- Software genérico.** Sistemas independientes que son desarrollados por una empresa de software y se venden en el mercado abierto a cualquier cliente que desee comprarlo. Ejemplos de estos productos son: procesadores de texto, hojas electrónicas, sistemas gestores de base de datos, paquetes para diseño gráfico, sistemas ERP, CRM, etc. [12, p. 6].
- Software personalizado o a la medida.** Sistemas que están destinados para un cliente en particular [12, p. 6].
- Satisfacción laboral.** Sentimiento positivo acerca de un puesto de trabajo que surge de la evaluación de sus características [30, p. 74].
- Servicio.** Es el resultado de llevar a cabo una actividad en la interfaz entre el proveedor y el cliente y generalmente es intangible. Un servicio puede ser una actividad realizada sobre un producto tangible entregado al cliente, una actividad realizada sobre un producto intangible entregado al

cliente, la entrega de un producto intangible, la creación de un ambiente para el cliente [1, p. 164]. Cualquier actividad primaria o complementaria que no produce directamente un bien tangible, es decir, la parte sin producto de la operación entre el comprador (cliente) y el vendedor (proveedor) [3, p. 58].

Subproceso. Partes definidas de un proceso. [1, p. 148].

Usabilidad. La facilidad con la que los usuarios pueden aprender a usar y usar el sistema [25, p. 465].

8 Siglas

AEC. Asociación Española para la Calidad.

AENOR. Asociación Española de Normalización y Certificación.

ASQ. *American Society for Quality* (Sociedad Americana para la Calidad).

CMMI. Capability Maturity Model Integration (Modelo de Madurez y Capacidad Integrado).

CMU. Carnegie Mellon University (Universidad Carnegie Mellon).

CRM. *Customer Relationship Management* (Sistema para la Gestión de la Relación con el Cliente).

CBTS. Computer Science and Technology Board (Buró de Ciencias de la Computación y Tecnología).

CSDP. Certified Software Development Professional (Profesional Certificado en el Desarrollo de Software).

EOQ. *European Organization for Quality* (Organización Europea para la Calidad).

ERP. *Enterprise Resource Planning* (Sistema para el Planeamiento de Recursos de la Empresa).

IEEE. *Institute of Electrical and Electronics Engineers* (Instituto de Ingenieros Eléctricos y Electrónicos).

IEEE-CS. *Institute of Electrical and Electronics Engineers' Computer Society* (Sociedad de Computación del Instituto de Ingenieros Eléctricos y Electrónicos).

MBTI. Myers-Briggs Type Indicator (Indicador de Tipo de Myers-Briggs).

ISO. *International Standardization Organization* (Organización Internacional de Normalización).

PCMM. People Capability Maturity Model (Modelo de Madurez y Capacidad para Personas).

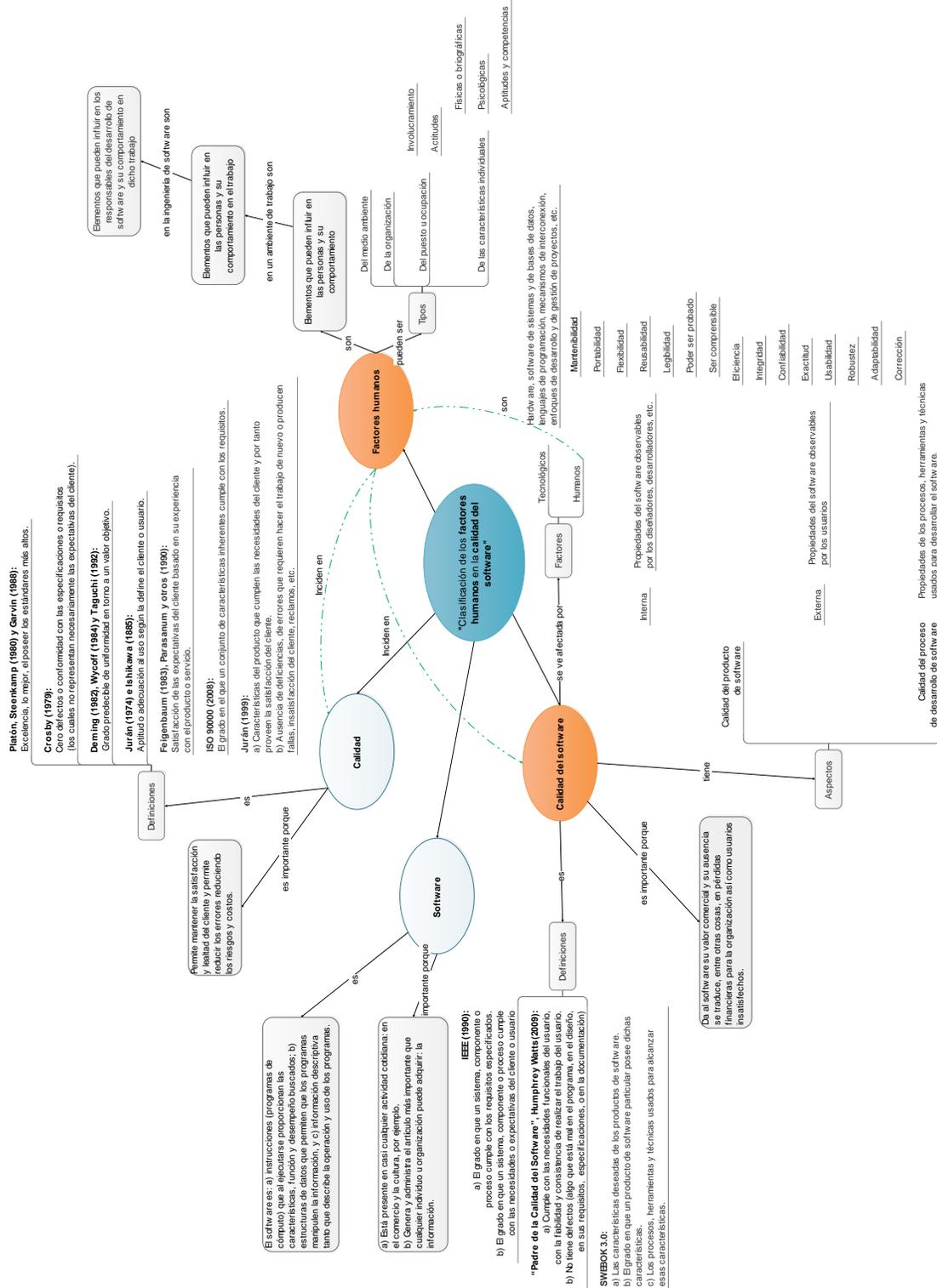
SEI. Software Engineering Institute (Instituto de Ingeniería de Software).

SWEBOK. *Software Engineering Book of Knowledge* (Libro del Conocimiento de la Ingeniería de Software).

UNE. Una Norma Española.

9 Anexos

9.1 Anexo No. 1 – Mapa conceptual



9.2 Anexo No. 2 – Matriz de congruencia metodológica

Tema de investigación		Variables		Propósito de la investigación									
"Factores humanos en la calidad del software".		Independiente: Factores humanos. Dependiente: Calidad del software. Intervinientes: Entorno organizacional y de trabajo (cultura organizacional, valores, nivel de madurez, estilo de dirección), características biográficas (sexo, edad, experiencia, discapacidades), etc.		La investigación permitirá comprender los factores humanos que investigaciones primarias establecen que inciden en la calidad del software, reconocer las buenas prácticas que posibilitan el desarrollo de software de calidad y algunos parámetros o indicadores para evaluar los factores humanos como parte de los procesos de control y aseguramiento de la calidad del software. La identificación y caracterización de los factores humanos que inciden en la calidad del software se hará inicialmente a partir de la revisión de la literatura. Luego, mediante una investigación de campo se contextualizarán o reconocerán los factores humanos con profesionales que trabajen o hayan trabajado en el desarrollo de software en El Salvador.									
Preguntas de investigación		Objetivo general	Objetivos específicos	Marco conceptual (1)	Marco metodológico (2)	Instrumentos	Muestra	Entregables					
1) ¿Qué factores humanos han sido estudiados en su relación con la calidad del software? Esta pregunta se responderá con la identificación y caracterización de los factores humanos que inciden en la calidad del software a partir de investigaciones primarias. 2) ¿Qué factores humanos en la calidad del software han sido más estudiados y cuales falta por investigar? Esta pregunta se responderá con la identificación y caracterización de los factores humanos que inciden en la calidad del software a partir de investigaciones primarias.		"Identificar y caracterizar los factores humanos que inciden en la calidad del software desde la perspectiva del control y aseguramiento de la calidad".	1) Identificar los factores humanos que, según la revisión de la literatura, inciden en la calidad del software.	Calidad del software (1.1) Factores humanos (1.2) Factores humanos en la calidad (1.3) Factores humanos en la calidad del software (1.4)	Revisión de la literatura (2.2) Investigación de campo (2.2) Mixta y secuencial (2.3)	Ficha de factores humanos (2.4) Ficha de categorías (2.4)	No aplica	1) Perfil del trabajo de graduación 2) Identificación de los factores humanos que inciden en la calidad del software					
3) ¿Qué buenas prácticas relacionadas con los factores humanos se identifican para posibilitar el desarrollo de software de calidad? Esta pregunta se responderá con la identificación de las buenas prácticas, relacionadas con los factores humanos identificados, que posibilitan el desarrollo de software de calidad.			2) Identificar las buenas prácticas, relacionadas con los factores humanos identificados, que posibilitan el desarrollo de software de calidad.										
4) ¿Qué parámetros o indicadores para evaluar los factores humanos se identifican y que podrían utilizarse en el control y aseguramiento de la calidad? Esta pregunta se responderá con la identificación de algunos parámetros o indicadores para evaluar los factores humanos como parte de los procesos de control y aseguramiento de la calidad del software.			3) Identificar algunos parámetros o indicadores, en los factores humanos identificados, que podrían utilizarse en el control y aseguramiento de la calidad del software.										
5) ¿Cómo podrían clasificarse los factores humanos en la calidad del software? Esta pregunta se responderá con una propuesta de clasificación de los factores humanos que inciden en la calidad del software.			4) Desarrollar una clasificación de los factores humanos identificados que inciden en la calidad del software.										
6) ¿Qué factores humanos que inciden en la calidad del software se reconocen en el contexto salvadoreño? Esta pregunta se responderá con el reconocimiento de los factores humanos que según la perspectiva y experiencia de profesionales en el desarrollo de software en El Salvador inciden en la calidad del software (objetivo específico 5).			5) Reconocer los factores humanos que según la perspectiva y experiencia de los profesionales en el desarrollo de software en El Salvador inciden en la calidad del software.						Factores humanos en la calidad (1.3) Factores humanos en la calidad del software (1.4)	Cualitativa (2.3) Fenomenológica (2.3) Entrevistas semiestructuradas (2.4) Análisis abierto y axial (2.5)	Guía de entrevista (2.4) Protocolo (2.4) Registro o bitácora (2.4)	No probabilística de expertos (2.6) 5 expertos	3) Resultados de la investigación de campo. 4) Tesina 5) Artículo
7) ¿Qué grado de importancia se le dan a los factores humanos en la calidad del software en el contexto salvadoreño? Esta pregunta se responderá con la determinación del grado de importancia que profesionales en el desarrollo de software en El Salvador dan a los factores humanos en la calidad del software.			6) Determinar el grado de importancia que los profesionales en el desarrollo de software en El Salvador dan a los factores humanos que inciden en la calidad del software.						Factores humanos en la calidad del software (1.4)	Cuantitativa (2.3) Descriptiva, no experimental (2.3) Cuestionarios (2.4) Análisis descriptivo (2.5)	Cuestionario (2.4) Protocolo (2.4)	No probabilística de expertos (2.6) 40 expertos	
Alcances			(1) Marco conceptual						(2) Marco metodológico				
1) Se identifican los factores humanos que se originan en las competencias y habilidades del recurso humano involucrado en el desarrollo de software, de sus relaciones con su entorno organizacional y de sus interacciones con otros individuos en el ambiente de trabajo. 2) Se consideran factores humanos relacionados con características individuales como la personalidad, el carácter y el estado de ánimo. 3) Se considera el factor humano relacionado con el entorno ambiental de trabajo. 4) No se consideran factores humanos relacionados con el entorno ambiental, el entorno social fuera del ambiente de trabajo y con características individuales como la edad, sexo, raza y discapacidades. 5) Se identifican inicialmente los factores humanos a partir de una revisión de la literatura sobre el tema. 6) Se identifican las buenas prácticas, algunos parámetros e indicadores 7) Se determina cuáles de esos factores humanos identificados son reconocidos por profesionales en El Salvador. 8) Se determina el grado de importancia que se le dan a los factores humanos en la calidad del software según la perspectiva y experiencia de los profesionales participantes en la investigación.		(1.1) Calidad del software: IEEE: Calidad del software es "a) el grado en que un sistema, componente o proceso cumple con los requisitos especificados, y b) el grado en que un sistema, componente o proceso cumple con las necesidades o expectativas del cliente o usuario" [21]. Humphrey Watts: Software de calidad es aquel que "a) cumple con las necesidades funcionales del usuario, con la fiabilidad y consistencia de realizar el trabajo del usuario, y b) no tiene defectos (algo que está mal en el programa, en el diseño, en sus requisitos, especificaciones, o en otra documentación)" [22]. (1.2) Factores humanos: Son todos aquellos elementos que pueden influir en las personas y su comportamiento. En un contexto de trabajo, factores humanos son los factores del medio ambiente, de la organización y de las interacciones con otros, y las características individuales que influyen en el comportamiento en el trabajo [29]. (1.3) Factores humanos en la calidad: ISO 9001: Identifica los factores humanos que se originan en las competencias, como: la educación, la formación, las habilidades y la experiencia [7, p. 7]. ISO 10018: Identifica otros factores asociados a las competencias, como: el reclutamiento, la educación, la formación, el empoderamiento, la responsabilidad, la conciencia de calidad, la creatividad e innovación. También identifica los factores humanos que se derivan de las interacciones de las personas con la organización: las líneas de responsabilidad, las líneas de autoridad y el liderazgo. Y finalmente identifica los factores que se originan en las relaciones con otros individuos el realizar el trabajo: el trabajo en equipo, la colaboración, la actitud, la motivación, la comunicación, el compromiso, las redes de relaciones y los reconocimientos [34]. (1.4) Factores humanos en la calidad del software: <ul style="list-style-type: none"> - McConnell [25, pp. 682, 820]: influencia de las competencias individuales en la calidad del software. - McConnell [25, p. 682]: relación de la experiencia y la calidad del código y la productividad. - Mathew citado por Lavallée y Robillard [36]: impacto de los factores organizacionales. - Nagappan, Murphy y Basili [37]: relación entre la estructura organizativa, la cohesión del equipo y el número de personas modificando el código con la calidad del software - Lavallée y Robillard [36]: identifica aspectos culturales que inciden negativamente en la calidad del software. - Jaktman citado por Lavallée y Robillard [36]: describe la relación entre los valores organizacionales y la calidad del software. - Prifling [38]: explora los estilos de liderazgo que promueven la calidad en proyectos de desarrollo. - Basili y Caldiera [39] proponen las "fábricas de experiencia" para mejorar la calidad del software. - Meneely y Williams [40] y Mahajan [41]: establecen una posible relación entre el tamaño del equipo y la calidad. - Bird, Nagappan, Murphy, Gall y Devanbu [42]: evalúa la tasa de errores en el desarrollo de Windows Vista en equipos distribuidos y equipos locales. 			(2.1) Enfoque de la investigación Se adhiere a las ideas de una visión filosófica constructivista. (2.2) Estrategia de investigación Comprende una revisión de la literatura y una investigación de campo . La revisión de la literatura permitirá enmarcar el problema y contextualizar el proceso de investigación; y la investigación de campo permitirá validar en el contexto salvadoreño los factores humanos identificados. (2.3) Tipo de investigación Se desarrollará una investigación mixta y secuencial . Investigación cualitativa en su primera fase donde se buscará entender el problema basándose en las opiniones y experiencias de cada uno de los participantes; e investigación cuantitativa en una segunda fase donde se buscará ampliar el entendimiento del problema con un grupo de participantes más amplio. Investigación fenomenológica para la fase cualitativa, y descriptiva no experimental para la fase cuantitativa. (2.4) Técnicas de recolección de datos Se utilizarán métodos cualitativos y cuantitativos según la fase. Para la revisión de la literatura se documentarán los hallazgos en una ficha de categorías de factores humanos y en una ficha de factores humanos. Para la investigación de campo, en la primera fase, la investigación cualitativa, se utilizará el método de la entrevista semiestructurada con preguntas abiertas para entender los factores humanos que inciden en la calidad del software. En la segunda fase, la investigación será cuantitativa, se utilizará un cuestionario que permitirá validar los factores humanos que inciden en la calidad del software con un grupo de participantes más amplio. (2.5) Técnicas de análisis de datos En la investigación cualitativa se recibirán de las entrevistas datos no estructurados, a los que el investigador deberá proporcionarles la estructura mediante un proceso de exploración, organización, descripción y descubrimiento. Para el análisis de datos se utilizarán métodos de codificación abierta o de primer nivel con el fin de comparar unidades y crear categorías; y métodos de codificación axial o de segundo nivel para comparar categorías. En la investigación cuantitativa se recibirán datos a partir de los cuestionarios que podrán ser analizados mediante métodos de análisis descriptivos. (2.6) Participantes de la investigación En ambas fases del estudio participarán profesionales que trabajen o hayan trabajado en el desarrollo de software en El Salvador. La caracterización de los participantes y la determinación del número de participantes será no probabilístico de expertos según los siguientes criterios: nivel de estudios, años de experiencia, y datos biográficos. (2.7) Proceso de investigación <ol style="list-style-type: none"> Inicio: comprende las actividades desde la concepción hasta la definición del tema y planeación del trabajo de campo. Ejecución: <ol style="list-style-type: none"> Revisión de la literatura: permitirá satisfacer los objetivos específicos 1 al 4 que competen a la identificación y caracterización de los factores humanos que inciden en la calidad del software, las buenas prácticas en esos factores humanos que posibilitan el desarrollo de software de calidad, los parámetros o indicadores y la clasificación. Investigación de campo: permitirá satisfacer los objetivos específicos 5 y 6 que competen al reconocimiento de los factores humanos en la calidad del software y su grado de importancia en el contexto salvadoreño. Seguimiento y control: comprende las actividades de seguimiento del proyecto de investigación. Cierre: comprende las actividades para el desarrollo de los informes finales y su presentación. 								
Justificación		Teórica: La calidad es acerca de las personas [5, p. 15]. Los factores humanos son el origen de la mayor parte de los problemas en el desarrollo de software [13, p. 7] [14, pp. 74-76]. Si el empleado desarrollando el software no tiene los conocimientos y habilidades apropiadas, o no tiene el interés, la calidad del software se verá afectada negativamente [31, p. 47]. Las principales causas de mala calidad tienen su origen en el factor humano [35]. Práctica y profesional: La identificación, clasificación y compilación de los factores humanos y de estudios que confirman su incidencia en la calidad proveerá de información para comprenderlos y reconocer las buenas prácticas que posibiliten el desarrollar software de calidad. Puede servir de apoyo en la evaluación del recurso humano, el entorno organizacional y de trabajo con el propósito de desarrollar software de calidad.											

9.3 Anexo No. 3 – Guía de entrevista

Guía de entrevista: *Factores humanos en la calidad del software*

A. Introducción

- 1) Presentación del entrevistador.
- 2) Descripción general de la investigación:
 - a. La investigación de campo tiene por objetivo: *“Reconocer los factores humanos que según la perspectiva y experiencia de profesionales en el desarrollo de software en El Salvador inciden en la calidad del software, y determinar su grado de importancia”*.
 - b. La investigación tiene 2 fases: una cualitativa (entrevistas) cuyo objetivo es explorar la problemática planteada por el tema de investigación y servir de insumo para la construcción del instrumento de la segunda fase; y una segunda fase que es cuantitativa (encuesta).
 - c. Los participantes del estudio son profesionales que hayan trabajado o trabajen en el desarrollo de software de El Salvador.
- 3) Características de la entrevista:
 - a. La entrevista es semiestructurada.
 - b. Tiene una duración aproximada de 30 a 45 minutos.
 - c. Se garantiza la confidencialidad del entrevistado.
 - d. Si es posible, se solicita permiso para grabar.

B. Preguntas para tipificar al entrevistado y la empresa

- 1) ¿Cuál es su edad?
- 2) ¿Cuál es su nivel educativo?
- 3) ¿En qué tipo de empresa es? ¿O no trabaja?
- 4) ¿Cuál es el sector económico de la empresa para que trabaja?
- 5) ¿Qué tamaño de empresa es?
- 6) ¿Cuál es su puesto de trabajo actual?
- 7) ¿Qué porcentaje del software que usan es desarrollado por la empresa? ¿Comprado?
- 8) ¿Qué porcentaje de los desarrolladores es fijo? ¿Subcontratado?
- 9) ¿Cuántos años de experiencia tiene en el desarrollo de software?
- 10) ¿Qué roles ha desempeñado en el desarrollo de software?
- 11) ¿Qué tamaños de equipo han tenido los proyectos en que ha participado?
- 12) ¿Qué duración han tenido los proyectos en que ha participado?

C. Preguntas para establecer el contexto del entrevistado:

Nota: el texto entre paréntesis en itálicas o al pie de página no debe mencionarse al entrevistado, únicamente es para indicar al entrevistador los conceptos/temas que el entrevistado debería estar mencionando y evaluar su conocimiento/práctica de los conceptos claves de la investigación.

- 1) ¿Cómo definiría "software de calidad"? ¿Qué atributos o características considera debe tener un software de calidad? ¹
- 2) ¿Tiene su empresa una unidad o responsable del control o aseguramiento de la calidad del software? ¿Qué responsabilidades tiene? ²
- 3) ¿Qué aspectos o atributos se evalúan en su empresa o evalúa usted para determinar la calidad del software? ³ ¿Tienen algún procedimiento formal de evaluación o una lista de verificación formal?
- 4) ¿Qué técnicas o herramientas se utilizan en su empresa o utiliza usted para evaluar o medir la calidad del software? ¿Qué métricas? ¿Qué métodos o niveles de prueba? ¿Qué herramientas de software? ¿Qué sistema de puntuación? ⁴

Preguntas de la investigación:

- 1) En su experiencia, ¿qué factores ha identificado inciden o contribuyen a la buena o mala calidad del software que se desarrolla? ⁵ ¿Por qué considera que inciden? ⁶ ¿Podría

¹ Según la ISO/IEC 9126, los atributos son:

Funcionalidad: idoneidad, exactitud, interoperabilidad, seguridad, conformidad con los requisitos funcionales.

Confiabilidad: madurez, tolerancia a fallas, habilidad para recuperarse.

Usabilidad: facilidad de entenderse, facilidad de aprenderse, facilidad de operarlo, facilidad de uso, ser atractivo.

Eficiencia: comportamiento respecto del tiempo, utilización de recursos.

Mantenibilidad: facilidad de ser analizado, facilidad de ser modificado, estabilidad, facilidad de ser probado.

Portabilidad: facilidad de adaptarse, facilidad de instalación, grado de coexistencia con otros sistemas, facilidad de ser reemplazado.

² **Software Quality Assurance (SQA):** la función de la calidad del software que asegura que los estándares, procesos y procedimientos son apropiados para el proyecto y son apropiadamente implementados.

Software Quality Control (SQC): la función de la calidad del software que verifica que el proyecto sigue sus estándares, procesos y procedimientos, y que el proyecto produce los productos con las características externas e internas requeridas. En el control de calidad todos los requisitos son "verificados" (¿Estamos construyendo correctamente el producto?) y "validados" (¿Estamos construyendo el producto correcto?), también las "pruebas" son parte del control de calidad, pero usualmente asociadas con pruebas de los requisitos funcionales.

³ **Aspectos de calidad estructural:** confiabilidad, eficiencia, seguridad, mantenibilidad, y tamaño adecuado.

Aspectos de calidad funcional: adecuación al uso, usabilidad, fallas o errores.

⁴ **Métodos de prueba:** estática/dinámica, caja blanca/caja negra.

Técnicas de prueba estática o verificación: revisiones de código, revisiones estructuradas, inspecciones.

Técnicas de prueba dinámicas o validación: pruebas (casos de prueba), usuarios finales.

Técnicas de prueba de caja blanca: pruebas de API, pruebas de cobertura de código (funciones, sentencias, decisiones), inyección de fallas, pruebas de mutación, análisis estático del programa.

Técnicas de prueba de caja negra: ECP (Equivalence Class Partitioning), análisis de valores límite, prueba de parámetros (All-pairs Testing), tablas de transición de estados, pruebas de tablas de decisión, evaluación fuzz, pruebas basadas en modelos, pruebas de casos de uso, pruebas exploratorias y pruebas basadas en la especificación.

Niveles de prueba: unitarias (de un componente o funcionalidad), de integración, de sistema, de interface, de aceptación de operación, de aceptación de usuario.

⁵ **Factor** es una circunstancia, hecho o influencia que contribuye a un resultado o consecuencia; algo que ayuda a producir o influenciar un resultado.

⁶ Factores humanos pueden ser:

- **Competencias:** capacidad de aplicar o usar un conjunto de conocimientos, destrezas y habilidades relacionadas para realizar con éxito las funciones o tareas de un trabajo.

compartir situaciones o experiencias que ilustren porque considera que esos factores inciden en la calidad del software?

2) ¿Cuáles de esos factores considera tienen origen humano? ⁷

-
- **Experiencia:** forma de conocimiento o habilidad derivados de la observación, participación y vivencia de un evento.
 - **Profesionalismo:** característica de la persona que desempeña un trabajo con pericia, aplicación, seriedad, honradez y eficacia.
 - **Empoderamiento:** la práctica de gestión de compartir información, recompensas, y poder con los empleados para que puedan tomar la iniciativa y tomar decisiones para resolver problemas y mejorar el servicio y el rendimiento.
 - **Conciencia de calidad:** comprender la pertinencia e importancia de las actividades que cada individuo realiza y de cómo estas contribuyen al logro de los objetivos de la calidad.
 - **Creatividad:** capacidad de generar nuevas ideas o conceptos, o de nuevas asociaciones entre ideas y conceptos conocidos, que habitualmente producen soluciones originales. Es sinónimo de pensamiento original, imaginación constructiva, pensamiento divergente o pensamiento creativo.
 - **Personalidad:** el conjunto de rasgos con las que una persona nace y que pueden ser afectados por su desarrollo social y cultural. Por ejemplo: extrovertido, tímido, enérgico, perezoso, seguro, divertido, negativo, etc.
 - **Carácter:** se refiere al conjunto de rasgos que definen la integridad de la persona, el conjunto de valores que tiene y que son aprendidos de otros (padres, maestros, amigos, etc.) Por ejemplo: bondadosa, honesta, leal, servicial, etc.
 - **Estado de ánimo:** actitud o disposición emocional en un momento determinado.
 - **Estructura organizativa:** disposición de las líneas de autoridad, comunicaciones, derechos y deberes que determina cómo se asignan los roles, el poder y las responsabilidades; como son controlados y coordinados; y cómo la información fluye entre los diferentes niveles de gestión.
 - **Cultura organizacional:** los conocimientos, creencias, valores, costumbres y otros hábitos adquiridos como miembro de la organización.
 - **Estilo de liderazgo:** la forma en que el líder dirige a un grupo de personas, proporciona indicaciones, las motiva e implementa planes. Estilos de liderazgo son: colaborativo o participativo, directivo, etc.
 - **Gestión del conocimiento:** identificar, usar y compartir de forma eficiente los activos intelectuales de una organización para mejorar su rendimiento y competitividad.
 - **Tamaño del equipo:** número de miembros en el equipo de desarrollo.
 - **Distribución del equipo:** la localización de los miembros del equipo de desarrollo. Por ejemplo: en el mismo lugar o distribuidos en diferentes lugares (oficinas, departamentos, países, continentes, etc.)
 - **Cohesión del equipo:** el grado en que los miembros del equipo contribuyen a la habilidad del grupo de funcionar como una unidad. Equipos cohesivos tienen vínculos emocionales y sociales que los vinculan unos a otros y les motiva por las metas comunes.
 - **Motivación:** la condición de estar entusiasmados por actuar o trabajar.
 - **Satisfacción:** el grado en que el individuo siente que sus necesidades (fisiológicas, de seguridad, sociales, de estima y de auto-realización) están satisfechas.
 - **Propiedad del código:** se refiere a distinguir quién trabajará en qué código, y a cuanto trabajo puede ser o debe ser compartido entre los miembros del equipo o diferentes equipos en la organización. Si hay alguien claramente responsable del código y de su calidad.
 - **Entorno ambiental:** los objetos y condiciones, tangibles y materiales que rodean el ambiente de trabajo.

⁷ Factores humanos pueden ser:

- **Competencias:** capacidad de aplicar o usar un conjunto de conocimientos, destrezas y habilidades relacionadas para realizar con éxito las funciones o tareas de un trabajo.
- **Experiencia:** forma de conocimiento o habilidad derivados de la observación, participación y vivencia de un evento.
- **Profesionalismo:** característica de la persona que desempeña un trabajo con pericia, aplicación, seriedad, honradez y eficacia.
- **Empoderamiento:** la práctica de gestión de compartir información, recompensas, y poder con los empleados para que puedan tomar la iniciativa y tomar decisiones para resolver problemas y mejorar el servicio y el rendimiento.

- 3) ¿Cuáles de esos factores son los 3 que considera tienen mayor grado de incidencia o impacto en la calidad del software? ¿Por qué le da ese grado de importancia?

Cierre

Dar las gracias e insistir en la confidencialidad y la posibilidad de su participación en la segunda fase de la investigación.

-
- **Conciencia de calidad:** comprender la pertinencia e importancia de las actividades que cada individuo realiza y de cómo estas contribuyen al logro de los objetivos de la calidad.
 - **Creatividad:** capacidad de generar nuevas ideas o conceptos, o de nuevas asociaciones entre ideas y conceptos conocidos, que habitualmente producen soluciones originales. Es sinónimo de pensamiento original, imaginación constructiva, pensamiento divergente o pensamiento creativo.
 - **Personalidad:** el conjunto de rasgos con las que una persona nace y que pueden ser afectados por su desarrollo social y cultural. Por ejemplo: extrovertido, tímido, enérgico, perezoso, seguro, divertido, negativo, etc.
 - **Carácter:** se refiere al conjunto de rasgos que definen la integridad de la persona, el conjunto de valores que tiene y que son aprendidos de otros (padres, maestros, amigos, etc.) Por ejemplo: bondadosa, honesta, leal, servicial, etc.
 - **Estado de ánimo:** actitud o disposición emocional en un momento determinado.
 - **Estructura organizativa:** disposición de las líneas de autoridad, comunicaciones, derechos y deberes que determina cómo se asignan los roles, el poder y las responsabilidades; como son controlados y coordinados; y cómo la información fluye entre los diferentes niveles de gestión.
 - **Cultura organizacional:** los conocimientos, creencias, valores, costumbres y otros hábitos adquiridos como miembro de la organización.
 - **Estilo de liderazgo:** la forma en que el líder dirige a un grupo de personas, proporciona indicaciones, las motiva e implementa planes. Estilos de liderazgo son: colaborativo o participativo, directivo, etc.
 - **Gestión del conocimiento:** identificar, usar y compartir de forma eficiente los activos intelectuales de una organización para mejorar su rendimiento y competitividad.
 - **Tamaño del equipo:** número de miembros en el equipo de desarrollo.
 - **Distribución del equipo:** la localización de los miembros del equipo de desarrollo. Por ejemplo: en el mismo lugar o distribuidos en diferentes lugares (oficinas, departamentos, países, continentes, etc.)
 - **Cohesión del equipo:** el grado en que los miembros del equipo contribuyen a la habilidad del grupo de funcionar como una unidad. Equipos cohesivos tienen vínculos emocionales y sociales que los vinculan unos a otros y les motiva por las metas comunes.
 - **Motivación:** la condición de estar entusiasmados por actuar o trabajar.
 - **Satisfacción:** el grado en que el individuo siente que sus necesidades (fisiológicas, de seguridad, sociales, de estima y de auto-realización) están satisfechas.
 - **Propiedad del código:** se refiere a distinguir quién trabajará en qué código, y a cuanto trabajo puede ser o debe ser compartido entre los miembros del equipo o diferentes equipos en la organización. Si hay alguien claramente responsable del código y de su calidad.
 - **Entorno ambiental:** los objetos y condiciones, tangibles y materiales que rodean el ambiente de trabajo.

9.4 Anexo No. 4 – Registro o bitácora de entrevista

Registro de entrevista: *Factores humanos en la calidad del software*

Fecha:		Hora de inicio/fin:	
Lugar:			
Identificación del entrevistado			
Sexo: <input type="checkbox"/> Femenino <input type="checkbox"/> Masculino		Nivel educativo: <input type="checkbox"/> Educación básica <input type="checkbox"/> Educación media <input type="checkbox"/> Bachillerato <input type="checkbox"/> Técnico <input type="checkbox"/> Licenciatura/Ingeniería <input type="checkbox"/> Maestría <input type="checkbox"/> Doctorado	
Edad:			
Identificación de la empresa			
Tipo de empresa: <input type="checkbox"/> No trabaja <input type="checkbox"/> Privada <input type="checkbox"/> Pública		Sector económico: <input type="checkbox"/> Primario <input type="checkbox"/> Secundario <input type="checkbox"/> Terciario	
Tamaño: <input type="checkbox"/> Micro (menos de 5 empleados) <input type="checkbox"/> Pequeña (entre 5 y 49 empleados) <input type="checkbox"/> Mediana (entre 50 y 99 empleados) <input type="checkbox"/> Grande (100 o más empleados)		Puesto de trabajo actual: <input type="checkbox"/> Director, gerente o jefe <input type="checkbox"/> Administrador <input type="checkbox"/> Operativo relacionado directamente con el desarrollo <input type="checkbox"/> Operativo no relacionado directamente con el desarrollo	
% de software desarrollado/ comprado:		% de desarrolladores fijos/ subcontratados:	
Experiencia del entrevistado en el desarrollo de software			
Años de experiencia en el desarrollo de software:			
Roles desempeñados: <input type="checkbox"/> Analista <input type="checkbox"/> Arquitecto <input type="checkbox"/> Diseñador <input type="checkbox"/> Programador <input type="checkbox"/> Integrador <input type="checkbox"/> Probador o verificador <input type="checkbox"/> Instalador <input type="checkbox"/> Administrador de proyecto		Tamaños de equipo: <input type="checkbox"/> 1 a 3 miembros <input type="checkbox"/> 4 a 6 miembros <input type="checkbox"/> 7 a 9 miembros <input type="checkbox"/> Más de 9 miembros	
		Duración de los proyectos: <input type="checkbox"/> Menos de 6 meses <input type="checkbox"/> 7 a 12 meses <input type="checkbox"/> 13 a 18 meses <input type="checkbox"/> 19 a 24 meses <input type="checkbox"/> Más de dos años	

Anotaciones:

9.5 **Anexo No. 5** – Cuestionario

Véase el documento [UDB_FHCS_Tesina_Anexo_5_Cuestionario.pdf](#) en el CD que acompaña el trabajo de graduación.

9.6 **Anexo No. 6** – Respuestas a la entrevistas

Véase el documento [UDB_FHCS_Tesina_Anexo_6_Respuestas_Entrevistas.pdf](#) en el CD que acompaña el trabajo de graduación.

9.7 **Anexo No. 7** – Respuestas al cuestionario

Véase el documento [UDB_FHCS_Tesina_Anexo_7_Respuestas_Cuestionarios.xlsx](#) en el CD que acompaña el trabajo de graduación.

9.8 **Anexo No. 8** – Resultados del cuestionario

Véase el documento [UDB_FHCS_Tesina_Anexo_8_Resultados_Cuestionarios.pdf](#) en el CD que acompaña el trabajo de graduación.