



UNIVERSIDAD NACIONAL DE LOMAS DE ZAMORA

FACULTAD DE INGENIERÍA

MEMORIA TESIS DOCTORAL

**Modelo de Trabajo Conceptual Referencial para
Ingeniería de Procesos.**

El caso de los Equipos de Desarrollo de Software

Doctorando: Marcelo Estayno

Directora: Dra. Claudia Minnaard

Co Directora: Dra. Marta Comoglio

Asesora: Esp. Fabiana Grinsztajn

Lomas de Zamora

9 noviembre de 2018

Agradecimientos

A Fabi (mi mujer, mi compañera, mi amiga, mi piloto de tormentas, mi socia de viajes), por la paciencia y el amor.

A Marta por las horas de trabajo, por el empujón por el aliento.

A Claudia, por su invaluable colaboración, por su trabajo.

A Oscar Pascal; por permitirme armar y conducir la Secretaría de Investigación de la Facultad durante 10 años, donde tuvimos muchos logros y confiar en mí.

A mis hijas y nietas, por ser mi motivo de lucha en la vida.

A mis padres por el ejemplo y amor.

RESUMEN

La investigación aborda el estudio de las metodologías que en la actualidad son utilizadas en el desarrollo del software, el estudio se focalizó en dos localidades maduras para este tipo de desarrollo industrial, en la República Argentina (Ciudad de Buenos Aires y Ciudad de Córdoba).

Para ello, se analiza la perspectiva de desarrolladores, identificando las competencias clave valoradas y analizando el papel que desempeña la comunicación y la trama vincular entre los stakeholders.

La finalidad es el diseño de un marco de trabajo y un conjunto de recomendaciones para buenas prácticas, a partir de la evaluación que surge de la investigación.

Para hacer el estudio, se analizaron antecedentes vinculados con la temática y se administró una encuesta.

Agradecimientos	II
RESUMEN	III
INDICE DE FIGURAS	VI
INDICE DE GRAFICOS	VI
INDICE DE TABLAS	VIII
CAPITULO I.....	1
Problema de investigación.....	1
Introducción.....	1
Principales interrogantes del trabajo	2
Hipótesis.....	3
Hipótesis 1:	3
Hipótesis 2:	3
Hipótesis 3:	3
Hipótesis 4:	3
Propósito del Trabajo de Investigación.....	3
Objetivo General	3
Objetivos específicos.....	3
Motivaciones para realizar la investigación.....	4
Valor de la Investigación	4
Preguntas específicas de la investigación	5
CAPÍTULO II.....	7
Software como Producto.....	7
Introducción.....	7
El software: cómo surge.....	7
Características del software.	8
Complejidad	9
Adaptabilidad	10
Modificabilidad	10
Invisibilidad	11
Evolución del Software.....	12
El software y su relación con el conocimiento.....	13
Resumen	15
CAPITULO III.....	16
Software como Proceso	16
Introducción.....	16
¿Cómo se construye el software?.....	16
Proceso de Desarrollo de software y ciclo de vida.....	16
Modelos en el Software y Modelos del Software.....	18

La importancia de modelar en el software.....	18
Notaciones y Lenguajes para modelar software.....	20
Procesos para el Desarrollo de Software.....	21
Procesos definidos – Proceso Unificado de Desarrollo	22
Las personas son decisivas	24
Procesos empíricos- Métodos Ágiles	24
El Proceso Grupal.....	32
El cono invertido	36
Resumen	38
CAPITULO IV	39
Industria del Software en Argentina	39
Composición del sector Informático en la República Argentina	39
Ventas e Ingresos desde Exterior del Sector SSI	42
Ventas por Actividad	44
Ingresos del Exterior por Actividad	45
Investigación, Desarrollo e Innovación	46
CAPÍTULO V	49
Perspectivas Metodológicas del Estudio.....	49
Alcances del Diseño	49
Análisis Descriptivo.....	49
Análisis Bivariado	50
Análisis Multivariado	52
Coeficiente de correlación Pearson (r).....	52
Análisis Factorial de Correspondencias Múltiples.....	52
Instrumento de Recolección de Datos para la Estrategia Cuantitativa	53
Definiciones operacionales de la Encuesta	53
Instrumento conceptual para la Encuesta : Sistema de Matrices de Datos	66
Focus Group.....	70
Características de los grupos focales	70
CAPITULO VI	71
Resultados.....	71
Análisis Descriptivo.....	71
Caracterización general de la muestra.....	71
Modalidades y Metodologías de Trabajo.....	77
Competencias y Formación requerida para integrar equipos eficientes	79
Análisis Bivariado	82
Análisis Multivariado	99
Análisis Factorial.....	102

Variable Tareas que se realizan	102
Variable Aspectos que conspiran con la obtención de un producto software de calidad	106
CAPITULO VII	110
Discusión y Presentación del Modelo	110
Principales Conclusiones	110
Propuesta de un modelo de trabajo conceptual referencial para los equipos de desarrollo de software	115
CAPITULO VII	125
Líneas a futuro	125
Bibliografía	126
ANEXOS	129
ANEXO I. Modelo de Encuesta	129
ANEXO II. Tablas de Frecuencia.....	132

INDICE DE FIGURAS

Figura 1. Evolución de la programación del sistema producto	8
Figura 2. Software como información	14
Figura 3. Las tres dimensiones críticas	21
Figura 4. Las 4 “P” en el desarrollo de software	24
Figura 5. Ciclo de Vida de Scrum.....	27
Figura 6. Ciclo de vida de XP.....	29
Figura 7. Relación entre las prácticas de XP	32
Figura 8. Esquema del cono invertido	37
Figura 9. Localización Centros SSI de la República Argentina.....	40

INDICE DE GRAFICOS

Gráfico 1. Evolución empresas SSI para el periodo 2006-2015	39
Gráfico 2. Empresas de Software en Argentina distribuidas por región.....	41
Gráfico 3. Tamaño de empresa en función de cantidad de empleados	42
Gráfico 4. Evolución anual de ventas totales (en millones de pesos corrientes) e Ingresos desde el exterior (en millones de pesos corrientes) 2008-2017	43
Gráfico 5. Evolución anual de ventas totales (en millones de USD) e ingresos desde el exterior (en millones de USD).....	43
Gráfico 6. Participación de las distintas actividades en el total de ventas – Año 2017	44
Gráfico 7. Principales clientes por sector y participación en las ventas. Año 2018	45
Gráfico 8. Participación de las actividades en los ingresos desde el exterior- 2018.....	46
Gráfico 9. Principales clientes por sector y participación en los ingresos del sector – 2018.....	46
Gráfico 10. I+D+i de empresas del sector correspondiente al año 2017.....	47
Gráfico 11. Principales objetivos de I+D+i – 2017.....	47
Gráfico 12. Relación entre las variables originales y los factores, así como su representación gráfica.	53
Gráfico 13. Edad de los encuestados.....	71
Gráfico 14. Actividad laboral	73
Gráfico 15. Sexo.....	73

Gráfico 16. Inserción en el ámbito laboral	74
Gráfico 17. Formación o título	75
Gráfico 18. Lugar de radicación laboral	76
Gráfico 19. Antigüedad en la última función	76
Gráfico 20. Forma de Trabajo	77
Gráfico 21. Utilización de Procesos Definidos	77
Gráfico 22. Metodologías basadas en procesos empíricos	78
Gráfico 23. Metodología de Gestión de Proyectos utilizadas	78
Gráfico 24. Competencias Técnicas requeridas	79
Gráfico 25. Competencias conductuales Requeridas.....	80
Gráfico 26. Formación más adecuada para integrar equipos de trabajo.....	80
Gráfico 27. Factores que dificultan la producción de software de calidad.....	81
Gráfico 28. Relaciones vinculares significativas para desarrollar software de calidad	82
Gráfico 29- Tabla de contingencia para las variables Edad y Antigüedad en la última función en el trabajo.....	83
Gráfico 30. Tabla de contingencia para las variables edad y Localización del Equipo de desarrollo en el que se desempeña.....	84
Gráfico 31. Tabla de contingencia para las variables antigüedad en la última función en el trabajo y Localización del Equipo de Desarrollo en el que se desempeña	85
Gráfico 32. Tabla de contingencia para las variables principales tareas que realiza en el trabajo y Localización del Equipo de Desarrollo en el que se Desempeña	86
Gráfico 33. Tabla de contingencia para las variables metodologías/frameworks basadas en filosofía Ágil y Localización del equipo de Desarrollo en el que se desempeña.....	88
Gráfico 34. Tabla de Contingencia para las variables, metodología de gestión de proyectos que utiliza el equipo de desarrollo y Localización del Equipo de desarrollo en el que se desempeña	89
Gráfico 35. Tabla de Contingencia para las variables Competencias Técnicas requeridas en los equipos de Localización del Equipo de Desarrollo en el que se desempeña.....	90
Gráfico 36. Tabla de Contingencia para las variables Tareas que Desarrolla y Sexo	91
Gráfico 37. Tabla de Contingencia para las variables Tareas que Desarrolla y utilización de metodologías basadas en procesos definidos	93
Gráfico 38. Tabla de Contingencia para las variables Tareas que Desarrolla y competencias conductuales requeridas	94
Gráfico 39. Tabla de contingencia para las variables, Tareas que desarrolla y aspectos de la relación vincular que es necesario mejorar.....	95
Gráfico 40. Tabla de contingencia para las variables competencias conductuales requeridas en los equipos de trabajo y competencias técnicas son requeridas	96
Gráfico 41. Tabla de Contingencia para las variables, competencias conductuales requeridas y aspectos de la relación vincular a mejorar.....	98
Gráfico 42. Tipologías a partir de competencias técnicas requeridas.....	101
Gráfico 43. Tareas que se desarrollan.....	103
Gráfico 44. Tipologías Análisis factorial para la variable tareas que se desarrollan.....	106
Gráfico 45. Aspectos que conspiran contra la calidad de los productos y o servicios de software	107
Gráfico 46. Tipología Análisis factorial para la variable factores que contribuyen a alcanzar productos de calidad.....	109

INDICE DE TABLAS

Tabla 1. Prueba de chi cuadrado de Independencia para variables edad y Antigüedad en la última función laboral	83
Tabla 2. Prueba de Chi cuadrado entre edad y Localización del Equipo de desarrollo en el que se desempeña.....	85
Tabla 3. Prueba de Chi Cuadrado para las variables antigüedad en la última función en el trabajo y localización del Equipo de Desarrollo en el que se desempeña.....	85
Tabla 4. Prueba de Chi cuadrado de independencia para las variables principales tareas que realiza en el trabajo y modalidad del equipo de desarrollo en el que trabaja	87
Tabla 5. Prueba de Chi Cuadrado correspondiente a las variables metodologías/frameworks basadas en filosofía Ágil y Localización del equipo de Desarrollo en el que se desempeña ...	88
Tabla 6. Prueba de Chi Cuadrado de independencia para las variables metodología de gestión de proyectos que utiliza el equipo de desarrollo y Localización del Equipo de desarrollo en el que se desempeña	89
Tabla 7. Prueba de Chi Cuadrado para las variables Competencias Técnicas requeridas en los equipos de Localización del Equipo de Desarrollo en el que se desempeña.....	91
Tabla 8. Prueba de Chi Cuadrado de independencia para las variables principales tareas que realiza y sexo.....	92
Tabla 9. Prueba Chi Cuadrado para las variables Tareas que Desarrolla y utilización de metodologías basadas en procesos definidos	93
Tabla 10. Principales tareas que desarrolla y competencias conductuales son requeridas	94
Tabla 11. Prueba de chi cuadrado para las variables Tareas que desarrolla y aspectos de la relación vincular que es necesario mejorar.....	96
Tabla 12. Prueba de chi cuadrado para las variables competencias conductuales requeridas en los equipos de trabajo y competencias técnicas son requeridas	97
Tabla 13. Prueba Chi cuadrado de Independencia para las variables, competencias conductuales requeridas y aspectos de la relación vincular a mejorar.	98
Tabla 14. Relaciones de Dependencia surgidas de la Prueba de Independencia Chi Cuadrado	99
Tabla 15. Estadísticas descriptivas para la variable Competencias Técnicas requeridas	100
Tabla 16. Matriz de correlación Pearson correspondiente a la variable Competencias Técnicas	100
Tabla 17. Correlaciones positivas y negativas para la variable Requerimientos Técnicos	101
Tabla 18. Estadísticas Descriptivas variable tareas que desarrolla	103
Tabla 19. Análisis Factorial. Tabla de Burt	105
Tabla 20. Estadísticas Descriptivas variable aspectos que por su ausencia o déficit conspiran contra software de calidad	107
Tabla 21. Factores que por su déficit o ausencia conspiran con la calidad del producto o servicio	108

CAPITULO I

Problema de investigación

Introducción

A lo largo de la historia del software se fueron planteando diferentes metodologías de desarrollo con el objeto de responder a las necesidades de los usuarios, en calidad, tiempo, costo y esfuerzo.

Las metodologías de desarrollo de software han definido lineamientos respecto de cómo los equipos de trabajo en las organizaciones conforman sus procesos de desarrollo. Estas metodologías se definen como “el conjunto de procedimientos utilizados para estructurar y controlar el proceso de desarrollo de sistemas de información”. Las mismas, consisten en una filosofía o paradigma de desarrollo que las sustenta y un conjunto de herramientas, técnicas y métodos que asisten al proceso de desarrollo de software.

Su avance ha acompañado la evolución de la industria y como consecuencia de la tecnología. El caso emblemático, hasta finales de los '80 era el paradigma estructurado (Yourdon, 1982), representado en lenguajes de programación como el COBOL, Visual Basic, Clipper, Fox, entre otros.

Desde el punto de vista metodológico el exponente principal del análisis de sistemas fue Edward Yourdon con su metodología *Análisis Estructurado Moderno* (Yourdon, 1982).

Con el surgimiento posterior del paradigma de orientación a objetos trascendieron lenguajes de programación como Smalltalk, C++, Java, C# y un amplio grupo de metodologías para el análisis y diseño de sistemas de información que seguían en esa línea de pensamiento. Las primeras metodologías reconocidas en la industria del paradigma orientado a objetos fueron: La metodología OMT (Rumbaugh, 1996), la metodología OOSE (Jacobson I. , 1994), y la metodología de Booch, (Booch G. , 1991).

Las organizaciones toman como base estas metodologías para crear los procesos de desarrollo que utilizan los grupos de trabajo. Estos procesos también han sufrido transformaciones y han evolucionado acompañando las necesidades del mercado de software. Se identifican dos tipos de procesos, los procesos de control definidos y los procesos de control empíricos. Las metodologías elegidas están fuertemente condicionadas por estos tipos de procesos y por consecuencia el comportamiento de los grupos de trabajo está supeditado a esa elección.

La disciplina informática ha logrado arribar a metodologías de desarrollo de software maduras en virtud de satisfacer las necesidades de los usuarios y la demanda creciente, incrementado la capacidad en

el desarrollo de software, respondiendo a necesidades de mayor producción y demanda. Estas metodologías, han mejorado la calidad y la productividad en los ambientes de desarrollo de software y es por esto por lo que podemos considerarlas metodologías maduras.

Sin embargo, aún se observan resistencias en la adopción de estas metodologías por parte de la industria en forma completa. Esta resistencia está sustentada en la suposición de que se incrementan los costos, o bien se subestiman sus beneficios; suponiendo o malinterpretando su aplicación, utilizando en su reemplazo metodologías más conocidas pero perimidas.

En el año 2001 se formalizó el surgimiento de un conjunto de metodologías basadas en procesos de control empíricos; que sus autores "Kent Beck; James Grenning; Robert C. Martin; Alistair Cockburn; Jeff Sutherland; Martin Fowler y otros acordaron en denominar "ágiles". Las metodologías ágiles se definen como un conjunto de lineamientos y mejores prácticas para el desarrollo de software que valoran tanto la propia experiencia como la de otros involucrados en el proceso de desarrollo.

El movimiento ágil no es anti-metodología, de hecho lo que busca es restaurar la credibilidad en la palabra metodología, considerándola como una herramienta que ayuda al proceso y no lo detiene (Agile Alliance, 2001).

Para muchas personas involucradas en el desarrollo de software el encanto de estas metodologías es su reacción ante la burocracia de las *metodologías tradicionales*, percibidas como "monumentales". Las metodologías ágiles buscan un equilibrio entre ningún proceso y demasiado proceso, proporcionando simplemente el proceso suficiente para que el esfuerzo valga la pena (Fowler, 2005).

Varios de los aspectos que aún las metodologías ágiles no terminan de resolver están vinculados a las relaciones interpersonales entre los diferentes stakeholders. Estos se definen como los grupos interesados y deben ser considerados un elemento esencial en la planificación estratégica de los negocios (Freeman E. , 1984); en la trama vincular (Pichon-Rivière, 2002), y en la comunicación.

Principales interrogantes del trabajo

Teniendo en cuenta las debilidades aún presentes en las metodologías ágiles se formulan los siguientes interrogantes:

¿Es posible conformar un marco de trabajo que permita mejorar las relaciones interpersonales entre los afectados en un proyecto de desarrollo de software?

¿Son las metodologías ágiles válidas para lograr los objetivos que se propone un equipo de trabajo durante el desarrollo de un producto software?

¿Los recursos con los que cuentan actualmente los equipos de trabajo son suficientes para tener relaciones interpersonales acordes a las exigencias a las que se ven sometidos?

¿Mejorar la comunicación en los equipos de trabajo, sería un aporte a los resultados esperados?

Hipótesis

En virtud de este último planteo, las siguientes hipótesis, funcionan como anticipaciones de sentido para el abordaje de la investigación.

Hipótesis 1:

La utilización de metodologías ágiles en el proceso de construcción de software mejora la calidad del producto resultante.

Hipótesis 2:

Un software que responde a las necesidades de los usuarios satisfactoriamente (un software con calidad), está relacionado con las competencias de los stakeholders, especialmente de los desarrolladores.

Hipótesis 3:

La comunicación entre los stakeholders es un factor que interviene de forma crítica en el desarrollo del software.

Hipótesis 4:

La trama vincular en el equipo de desarrollo, incluyendo al dueño del producto (Producto Owner del Scrum) y a los usuarios directos es un factor determinante de la calidad del producto a desarrollar.

Propósito del Trabajo de Investigación

Objetivo General

Proponer un framework (marco de trabajo) con recomendaciones y buenas prácticas, dirigida a mejorar las relaciones interpersonales y la comunicación entre los stakeholders de un proyecto de desarrollo de software.

Objetivos específicos

- Indagar cuáles son los métodos de desarrollo utilizados en el mercado, en las localidades maduras de la República Argentina.
- Identificar las competencias clave valoradas por los desarrolladores de software de los centros de desarrollo a investigar.
- Analizar el papel que desempeña la comunicación entre los stakeholders y la dinámica de los vínculos que se establecen en el equipo de desarrollo.

- Diseñar un modelo que se constituya en un marco orientado a definir un conjunto de buenas prácticas conducentes a mejorar el ecosistema de los equipos de desarrollo de software en función del análisis que surge de la investigación.

Motivaciones para realizar la investigación

Como Ingeniero en Sistemas de Información he tenido la oportunidad desde el año 1983 de participar y desempeñarme en distintos roles en equipos de desarrolladores de software: programador, analista, líder, arquitecto, implementador, capacitador, usuario, entre otras funciones. Esta experiencia personal me ha permitido observar el papel que las relaciones interpersonales y comunicacionales desempeñan y como éstas influyen en la calidad de los productos.

Como profesor universitario y complementado con la formación de psicólogo social, he sido un observador agudo de los aspectos más diversos de la profesión. Siempre ocupado en las competencias, o sea en las capacidades y habilidades de los profesionales de la informática para dar respuesta a la industria y a los requerimientos de la formación permanente de nuevos profesionales. También, me he desempeñado como formador de técnicos, sobre todo de técnicos en programación y de analistas funcionales, los dos perfiles más requeridos dentro de puestos laborales.

Por tal motivo, esta experiencia constituye una oportunidad para indagar a través de un proceso sistemático los ambientes de trabajo en la industria del software, que ponga a prueba las hipótesis planteadas y finalmente diseñar el modelo propuesto como propósito de este trabajo de tesis.

Finalmente se aspira a que a través del modelo propuesto, los ámbitos de trabajo en los que se adopte se contribuya a la mejora de la calidad de vida de las personas que desarrollan software y por consiguiente la mejora en la calidad de los productos, como así también a que la industria del software dé respuesta a las necesidades de la sociedad.

Valor de la Investigación

Varios son los sectores interesados en los aportes que pueden obtenerse como resultados de la investigación; a continuación se mencionan algunos de los que se consideran más relevantes: la industria, el gobierno y la academia.

Por un lado, se destaca como beneficiarios directos en la industria a los equipos de trabajo que se conforman cada vez que inicia un proyecto para desarrollar un producto de software, a estos equipos está especialmente dirigido el resultado de este trabajo.

La industria, por otra parte busca permanentemente nuevas metodologías para mejorar su productividad y calidad, en caso de incorporar una nueva metodología cuya adaptación resulte un incremento fácilmente incorporable.

Los lazos que se conforman entre la industria, cada una de sus organizaciones y los equipos de trabajo, eslabón final en esta cadena de valor; tienen visibles necesidades de ser mejorados especialmente en el campo de las relaciones comunicacionales.

Por su parte, los gobiernos, en su permanente búsqueda por favorecer el crecimiento de la industria, alientan las mejoras en los procesos productivos. Cabe aclarar, que en este caso, la industria del software es una actividad productiva, basada fundamentalmente en el conocimiento. En Argentina, se viene impulsando, por reconocerla como potencial del desarrollo, es así como existe una ley de Promoción de la Industria del software (Ley 26.692/11) que desgrava impositivamente a la industria. Entre los requerimientos para la aplicación solicita a las empresas, -entre otros requisitos-, la realización de actividades de investigación e innovación. El aporte de nuestro trabajo se orienta a contribuir en esa línea.

La academia en su búsqueda permanente de optimizar los planes de estudio de sus carreras siempre está atenta a la incorporación de nuevos contenidos curriculares que mejoren las competencias de los futuros profesionales. Es una ocupación continua de la academia satisfacer las necesidades del medio, en este caso corporizada por las organizaciones a las que se transfieren los graduados; donde éstos deben desempeñarse con las mejores y más actualizadas habilidades y capacidades.

Los resultados de la investigación benefician además a los profesionales del medio, insertos en los equipos de trabajo, que desarrollan su actividad en las organizaciones o en este tipo de industria donde hay grupos independientes de emprendedores, creando los productos software para el mercado global. Para ellos es especialmente útil mejorar su performance dando respuesta a las exigencias de calidad que les es demandada para lograr su desarrollo y evolución profesional.

Preguntas específicas de la investigación

Para delimitar nuestro campo de indagación, nos formulamos las siguientes preguntas:

¿Qué metodologías de desarrollo se utilizan en la industria del software en las localidades maduras?

¿Qué competencias conductuales y técnicas, tienen los integrantes de los grupos de desarrollo de software?

¿Cuáles son las competencias valoradas por los integrantes de los grupos de desarrollo de software?

¿Cómo se gestionan los proyectos de desarrollo de software?

¿Qué formación es requerida para los integrantes de un equipo de desarrollo de software?

¿Qué metodologías se utilizan para el desarrollo de software y cómo son valoradas?

¿Qué importancia tiene la comunicación en los procesos de desarrollo?

¿Qué conocemos de la trama vincular que se genera entre los involucrados y cuánto se trabaja en ella?

¿Cuáles son los aspectos que conspiran contra la obtención de un producto software de calidad?

CAPÍTULO II

Software como Producto

Introducción

El software en su origen era la parte insignificante del hardware, lo que venía como añadidura, casi como regalo. Sin embargo, no pasó mucho tiempo de su evolución y se generó una idea de software = programa, y en breve tiempo adquirió una entidad propia, la cual desde el comienzo fue acompañada de complejidad, por ser un producto difícil de entender, difícil de explicar, difícil de construir y difícil de mantener, o sea un producto de naturaleza compleja, intangible, maleable y con diversas representaciones, en constante cambio y evolución. En este sentido, se considera que el software es conocimiento (Freeman P., 1987) es información y que en su evolución tiene distintos niveles de abstracción.

El software como producto, es el resultado de la ejecución de un proyecto que usa procesos de desarrollo, (Jacobson I. B., 2000). Asimismo, se ha convertido en ubicuo (Evans, 2004); está inserto en la mayoría de las actividades de la persona humana, en su cotidianeidad.

Los aspectos señalados precedentemente, permiten tener una idea de la complejidad que su comprensión implica, motivo por el cual es importante no subestimar ni simplificar su abordaje.

El software: cómo surge.

En el diccionario bilingüe Longman, se define el término software como el conjunto de programas que controlan la operación de una computadora. (Pressman, 2005) considera que el software se forma con:

1) las instrucciones (programas de computadoras) que al ejecutarse proporcionan las características, funciones y el grado de desempeño.

2) las estructuras de datos que permiten que los programas manipulen la información de manera adecuada.

3) los documentos que describen la operación y el uso de los programas.

Sommerville, (Sommerville, 2011) define al software de manera similar: *“programas de cómputo y documentación asociada. Los productos de software se desarrollan para un cliente en particular o para un mercado en general”*. Por su parte, Freeman, caracteriza al software en los siguientes términos:

“El alma y cerebro de la computadora, la corporización de las funciones de un sistema, el conocimiento capturado acerca de un área de aplicación, la colección de los programas, y los datos necesarios para convertir a una computadora en una máquina de propósito especial diseñada para

una aplicación particular, y toda la información producida durante el desarrollo de un producto de software” (Freeman P. , 1987)

El software no sólo hace referencia a los programas ejecutables sino también representa diversas cosas, las cuales difieren en el tiempo, con las personas y principalmente en cómo se lo va a emplear (Freeman P. , 1987). El software viabiliza el producto más importante de nuestro tiempo: *la información*.

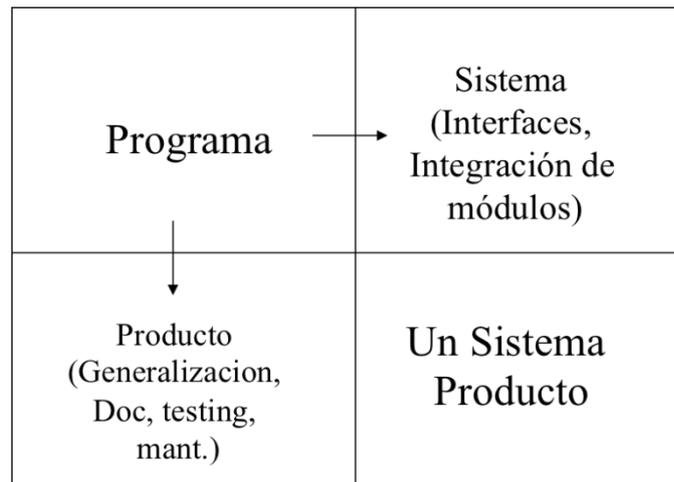


Figura 1. Evolución de la programación del sistema producto
Fuente Freeman, 1987

Características del software.

El software es un elemento lógico, no físico, para comprender mejor al software es importante conocer las características que lo distinguen de otros productos que construye el hombre. Se puede analizar al software desde diversos puntos de vista y desde allí establecer sus propiedades. En (Brooks, 1995), edición aniversario de su libro en el que incluye su reconocido artículo “No Silver Bullet – Essence and Accident in Software Engineering”, el autor plantea que hay dificultades que son inherentes a la naturaleza del software y otras que son accidentes que están en su producción.

Las características esenciales se agrupan en *complejidad, conformidad, cambio e invisibilidad*.

Siguiendo a Aristóteles, Brooks divide las características del software en esenciales y accidentales. Las esenciales son dificultades inherentes a la naturaleza del software, las accidentales son aquellas dificultades temporales que ahora existen en su producción, pero que no son inherentes y por tanto se pueden evitar y pueden variar a lo largo del tiempo.

La esencia de una entidad de software es una construcción de conceptos interrelacionados, esta esencia es abstracta en el sentido de que la construcción conceptual es la misma bajo muchas representaciones, no obstante, es altamente precisa y ricamente detallada.

Lo difícil de hacer software es la especificación, diseño y prueba de esta *construcción conceptual*, no el trabajo de representarla y verificar la fidelidad de la representación. En este sentido se puede afirmar que los mayores errores que cometemos son los conceptuales.

Los errores derivados de la creencia, de que se puede hacer una simplificación extrema de la realidad, logrando que algo del mundo real, que es difícil, se convierta por arte de magia en relativamente fácil dentro del software. *Si este postulado es cierto, la construcción del software será siempre compleja.*

Consideremos ahora las siguientes cuatro propiedades de esta esencia irreductible del software:

Complejidad

Las entidades de software son más complejas por su tamaño que tal vez cualquiera otra obra humana, no hay dos partes iguales. Para el caso de que las haya, de las dos partes similares se realiza un subprograma, ya sea abierto o cerrado. En ese sentido, los sistemas de software difieren profundamente de las computadoras, edificios o automóviles, en donde abundan los elementos repetidos.

De igual manera, la construcción de software a gran escala no es simplemente una repetición de los mismos elementos en tamaños o en cantidades más grandes, sino que se necesita aumentar el número de elementos diferentes. Casi siempre los elementos interactúan en alguna forma no lineal y la complejidad del todo aumenta exponencialmente.

Muchos de los problemas clásicos del desarrollo de software se derivan de esta complejidad esencial y de su aumento no lineal en el tamaño. De la complejidad viene la dificultad de comunicación entre los miembros del equipo, que lleva a deficiencias del producto, exceso de costo y atrasos entre otros aspectos.

También de la complejidad, se deriva la dificultad de enumerar todos los estados posibles de un programa; y de ahí se traduce en escasa confiabilidad.

También de la complejidad de una función deviene la complejidad de invocar la función, lo que hace a los programas difíciles de usar. Asimismo, de la complejidad de la estructura deriva la dificultad de extender los programas con nuevas funciones sin crear efectos laterales. Por último, de la complejidad de la estructura se llega a estados no visualizados que constituyen problemas de seguridad.

Cabe advertir, que de la complejidad no sólo se derivan problemas técnicos, sino también administrativos, lo que dificulta las revisiones, impidiendo la integridad conceptual. En particular, se hace difícil encontrar y controlar todos los cabos sueltos que conlleva un problema de aprendizaje y entendimiento, que impacta negativamente en los equipos de trabajo ante los cambios derivados de la rotación de personal.

Adaptabilidad

Los agentes vinculados al software no son los únicos que encaran la complejidad. Se puede observar que la física manipula objetos sumamente complejos al nivel "fundamental" de las partículas, pero, sin embargo, los físicos trabajan sobre la firme base de la existencia de principios unificadores, ya sea en quarks o en teorías de campo unificadas. En este sentido Einstein, afirmaba que debe haber explicaciones simples de la naturaleza, porque "Dios no es caprichoso o arbitrario".

Ninguna fe similar apoya al ingeniero de software. No cabe duda de que la mayoría de la complejidad que él debe manejar es arbitraria, forzada sin razón alguna por la multiplicidad de instituciones y sistemas humanos cuyas interfaces deben ser respetadas. Ellas difieren de interfaz en interfaz y cambian con el tiempo, no se trata de exclusivas razones a necesidad, sino fundamentalmente a que fueron diseñadas por diferentes personas.

En muchos casos, el software debe adaptarse porque es el último en llegar y es percibido como el más adaptable. Pero en todos los casos la adaptación a otras interfaces crea mucha complejidad, la que no se puede eliminar solo rediseñando el software.

Modificabilidad

La entidad software está constantemente bajo presión para cambiar, circunstancia que también puede aplicarse a edificios, autos y computadoras. Sin embargo, se observa que aquellos objetos, una vez que son diseñados y armados, raramente se modifican después de manufacturados. O bien se reemplazan por modelos nuevos, o bien cambios esenciales se incorporan en las nuevas copias del mismo diseño básico. En este sentido, las modificaciones de los automóviles ya vendidos son muy poco frecuentes; los cambios en las computadoras son un poco más frecuentes. Pero en ambos casos, son mucho menos frecuentes que las modificaciones del software ya instalado.

Esta peculiaridad, es en parte porque el software de un sistema incluye su función, y la función es la que siente más las presiones de cambio. Esta característica, resulta inherente al software, porque se puede modificar fácilmente, ya que es un producto en materia de pensamiento puro, infinitamente maleable. Si bien, se puede observar que los edificios también se modifican, pero el alto costo de las modificaciones, entendido por todos, sirve para amortiguar los ánimos de quienes quieren cambios.

Todo software exitoso se cambia, y en este cambio se observan dos procesos simultáneos. Primero, en la medida en que el software prueba ser útil, la gente usa casos que estaban en el límite o más allá del campo de aplicación inicial. Las presiones para extender la funcionalidad vienen principalmente de los usuarios que les gusta la funcionalidad básica e inventan nuevos usos para ella. Segundo, el software exitoso sobrevive la vida normal de la máquina para la cual fue escrito. Si no son nuevas computadoras, por lo menos son nuevos discos, nuevas pantallas, nuevas impresoras; y el software debe adaptarse a estos nuevos vehículos.

En resumen, el producto de software está inmerso en una matriz cultural de aplicaciones, usuarios, leyes y máquinas en los cuales está embebido. Todos estos elementos cambian continuamente y sus cambios inexorablemente fuerzan modificaciones al producto de software.

Invisibilidad

Otra características del software es que es invisible y por lo tanto no visualizable.

Las abstracciones geométricas son herramientas poderosas, por ejemplo, el plano de planta de un edificio le ayuda al arquitecto y al cliente a evaluar espacios, flujos de tránsito, vistas, en tanto que en estos casos, las contradicciones y omisiones se hacen evidentes. Los dibujos a escala de partes mecánicas y los modelos de palitos para las moléculas, si bien son abstracciones, sirven para los mismos fines. Una realidad geométrica se captura con una abstracción geométrica.

En tanto, la realidad del software es inherentemente no espacial, por lo tanto no hay representaciones geométricas claras, del modo que la tierra tiene mapas, los circuitos de silicio tienen diagramas, las computadoras tienen esquemas de conexión.

Apenas intentamos hacer un diagrama de la estructura del software, se advierte que no constituye uno, sino muchos grafos dirigidos, generales, superpuestos. Estos grafos pueden representar el flujo de control, el flujo de datos, dependencias, secuencias de tiempo, relaciones nombre-espacio. Estos grafos ni siquiera son planos y mucho menos jerárquicos. De hecho, una manera de obtener control conceptual sobre la estructura es el corte forzoso de arcos hasta que uno o más de los grafos sean jerárquicos.

A pesar del progreso en las restricciones y simplificaciones de las estructuras del software, por la característica señalada “inherentemente no visualizables” impiden usar una de las herramientas conceptuales más poderosas de la mente.

Otro elemento para tener en cuenta es que “*se hace software con software*”. Esto no sólo dificulta el proceso de diseño en nuestra mente, sino que dificulta severamente la comunicación entre las mentes.

Robert Cochram, citado en (McConnell S. , 2001) utiliza las siguientes características para describir lo que es único y especial en el software:

- El software es intangible.
- Tiene alto contenido intelectual.
- Generalmente no es reconocido como un activo por los Contadores, por lo que no está en los balances.
- Su proceso de desarrollo es humano intensivo, basado en equipos y construido en proyectos.

- El software no exhibe una separación profunda entre I&D¹ y producción.
- El software puede ser potencialmente modificado, en forma permanente.

Pressman, (Pressman, 2005) expresa las características del software haciendo énfasis en la diferencia que éste tiene con el hardware:

- El software no se desarrolla, se manufactura.
- El software no se desgasta.
- A pesar de la tendencia de la industria a desarrollar por componentes, gran parte del software aún se construye a medida.

Pese a la diversidad de opiniones que expresan los autores respecto a las características del software, todos coinciden en su complejidad. Según Basili, (Basili V. , 1993) el software es intangible y su intangibilidad es lo que contribuye fuertemente a su complejidad. El autor señala que el software es complejo, en diversas dimensiones: complejo de construir, complejo de entender, complejo de explicárselo a otros.

Esta característica inherente del software, junto al hecho de que el software es, desarrollado y no construido, por un lado, y la falta de principios y componentes bien definidos, por el otro, hace de él un producto totalmente diferente a otros.

La existencia de requerimientos de cambio y evolución constante de su función y/o estructura, su desarrollo propenso a errores, la dificultad para su estimación, la falta de comprensión de las implicancias que trae aparejado el cambio, son algunas de las razones que confirman su complejidad.

Evolución del Software

El software evoluciona sencillamente porque se debe adaptar a los cambios permanentes del entorno, sean funcionales (exigencias de usuarios), operativos, de plataforma o arquitectura de hardware.

La dinámica de evolución del software es el estudio de los cambios del sistema. La mayor contribución en esta área fue realizada por Meir M. Lehman y Belady. Su trabajo continuó en la década de 1990, con Lehman y otros investigadores de relevancia en la realimentación en los procesos de evolución (M., 2009). A partir de esos estudios se propusieron un conjunto de leyes (conocidas como leyes de Lehman) respecto de los cambios producidos en los sistemas. Estas leyes son ampliamente aplicables.

Las Leyes de la *teoría unificada de la evolución del Software* son, las siguientes:

¹I&D: Investigación y Desarrollo

- *Cambio continuo*: Un programa que se usa en un entorno real necesariamente debe cambiar o se volverá progresivamente menos útil en ese entorno.
- *Complejidad creciente*: A medida que un programa en evolución cambia, su estructura tiende a ser cada vez más compleja. Se deben dedicar recursos extras para preservar y simplificar la estructura.
- *Evolución prolongada del programa*: La evolución de los programas es un proceso auto regulativo. Los atributos de los sistemas, tales como tamaño, tiempo entre entregas y la cantidad de errores documentados son aproximadamente invariantes para cada entrega del sistema.
- *Estabilidad organizacional*: Durante el tiempo de vida de un programa, su velocidad de desarrollo es aproximadamente constante e independiente de los recursos dedicados al desarrollo del sistema.
- *Conservación de la familiaridad*: Durante el tiempo de vida de un sistema, el cambio incremental en cada entrega es aproximadamente constante.
- *Crecimiento continuado*: La funcionalidad ofrecida por los sistemas tiene que crecer continuamente para mantener la satisfacción de los usuarios.
- *Decremento de la calidad*: La calidad de los sistemas de software comenzará a disminuir a menos que dichos sistemas se adapten a los cambios de su entorno de funcionamiento.
- *Realimentación del sistema*: Los procesos de evolución incorporan sistemas de realimentación multiagente y multibucle y estos deben ser tratados como sistemas de realimentación para lograr una mejora significativa del producto.

El software y su relación con el conocimiento

El software refleja las estructuras de conocimiento que existen en el mundo real o dominio del problema² que representa. En este contexto, *“es importante reconocer que la característica principal del software es que es esencialmente conocimiento empaquetado”*. (Freeman P. , 1987).

En (Freeman P. , 1987) se definen tres clases de información que intervienen en la producción del software.

La primera de estas categorías llamada *representaciones del software*, incluye a los programas, diseños detallados, representaciones de diseño, especificaciones de procesos y requerimientos, entre otras. En resumen, cualquier información que en alguna forma directa representa un conjunto de programas y sus datos asociados puede incluirse en esta categoría. La definición de un conjunto de representaciones consistente y la especificación de cómo lograr esa consistencia es uno de los principales problemas que

²Dominio del Problema es la parte del mundo donde el software debe producir los efectos deseados, junto con los medios disponibles para producirlos directa o indirectamente.

enfrentan quienes deben desarrollar software. Esta clase de información hace referencia al *software como producto*.

Otra clase de información establecida por Freeman es el *conocimiento de la Ingeniería de Software*. Esta categoría se refiere a toda la información relacionada con el proyecto, la tecnología de software, el conocimiento de sistemas similares, es decir cómo usar un método determinado o la información relacionada con un desarrollo específico (el cronograma creado para un proyecto).

La tercera clase de información, esencial para la creación del software es el *conocimiento específico del dominio*, incluye al entendimiento de un dominio determinado a ser desarrollado. Los ejemplos de esta clase de información son innumerables, la comprensión de un proceso físico, reglas de contabilidad, la atención médica en un hospital, etcétera. Mientras que esta información es claramente esencial para la creación del software, explicarla, presentarla en una forma útil es competencia de especialistas de cada dominio, ingeniero de control de procesos, contador o médico.

La Figura 2 muestra la relación entre las clases de información indicando que el espacio de coincidencia o superposición es pequeño. Esta intersección es crítica para la creación exitosa del software dado que contiene conceptualmente toda la información necesaria para la existencia del sistema (software como producto) deseado.

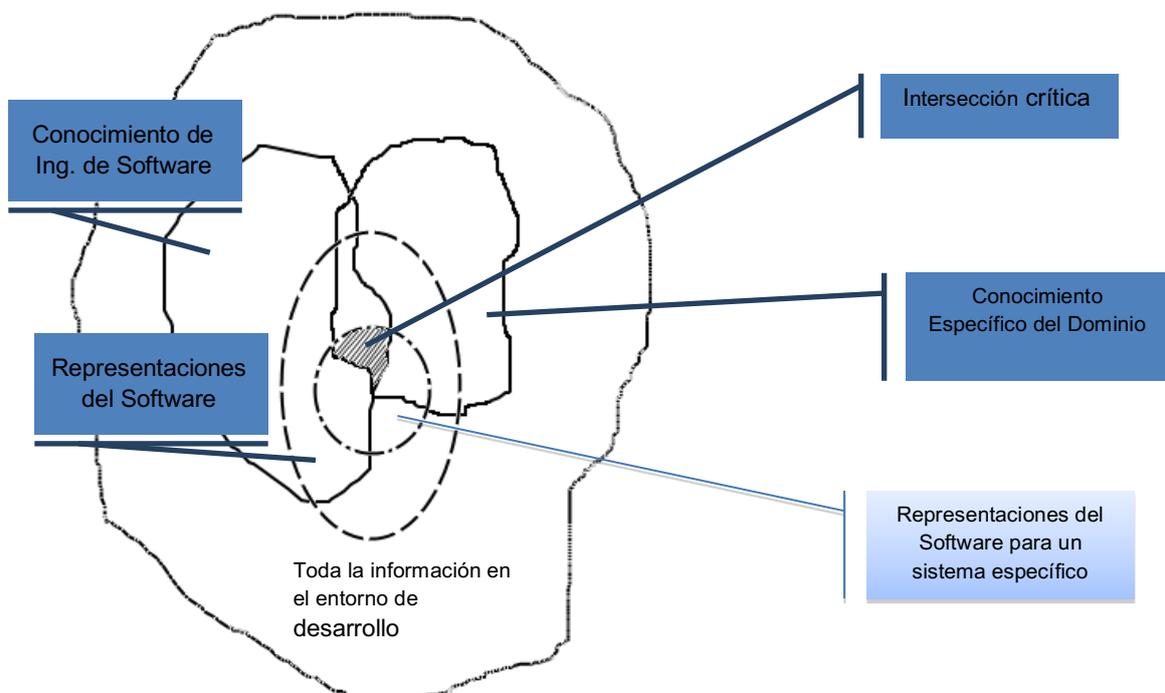


Figura 2. Software como información
Fuente (Freeman P. , 1987)

Resumen

En este capítulo se introdujo la problemática del software y su complejidad, que se evidencia desde el momento mismo que se necesita dar una definición acerca de él. Se describieron las características inherentes al software su relación con el conocimiento y las formas en las que se lo puede representar. Se presentó al software como producto en constante cambio y evolución.

Se trabajó alrededor de las ideas fuerza que se destacan a continuación:

- El software viabiliza el producto más importante de nuestro tiempo: *la información*.
- El software es intangible y su intangibilidad es lo que contribuye fuertemente a su complejidad.
- El software es complejo, complejo de construir, complejo de entender, complejo de explicárselo a otros.

El software es esencialmente conocimiento empaquetado. Debido a que el software es conocimiento materializado y dado que el conocimiento es en un comienzo, disperso, tácito (NONAKA, 1995) latente y en gran medida incompleto, el desarrollo de software es un proceso de aprendizaje social.

CAPITULO III

Software como Proceso

Introducción

¿Cómo se construye el software?

Debido a que el software es conocimiento materializado y dado que el conocimiento es en un comienzo, disperso, tácito, latente y en gran medida incompleto, el desarrollo de software es un proceso de aprendizaje social. El proceso proporciona una interacción entre los involucrados³ (Freeman E. , 1984) (usuarios, diseñadores, desarrolladores y analistas entre otros) y sus herramientas que por medio de la comunicación evolucionan y logran obtener más conocimiento útil que les permita alcanzar sus objetivos. (Sommerville, 2011)

Del análisis de las características del software se puede deducir que la construcción del software se realiza trabajando en equipo y utilizando proyectos.

En este capítulo se describirá en términos generales los conceptos que se utilizan a la hora de plantear un proyecto que obtendrá como resultado un producto de software.

Proceso de Desarrollo de software y ciclo de vida

El proceso software (Software Engineering Institute, 2010), incluye metodologías para el desarrollo de este, dado que las metodologías definen conceptos y herramientas para su producción, mientras que el proceso incluye además aspectos vinculados a la administración y organización del proyecto.

Cuando se trabaja para construir un producto de software, es importante seguir una serie de pasos predecibles, una especie de mapa de ruta, que ayude a crear un resultado de calidad y a tiempo. Ese mapa de ruta que debe seguirse se llama proceso de software, (Sommerville, 2011) .

Jacobson (Jacobson I. B., 2000), en un sentido muy general plantea que *“un proceso define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo”*, constituyendo así un marco de trabajo de las tareas a realizar (Pressman, 2005).

³Involucrado: del término en inglés “stakeholder” que Freeman define como “aquellos grupos que pueden afectar o ser afectados por el logro de los propósitos de la organización”. Pueden distinguirse varios tipos de stakeholders, según sea su influencia directa o indirecta sobre la empresa: los ‘primarios’ o ‘definicionales’ y los stakeholders ‘instrumentales’. Los primarios son vitales para el crecimiento continuo y sobrevivencia de cualquier empresa, mientras los stakeholders instrumentales están en el entorno amplio de la empresa y son aquellos que pueden influenciar a los primarios (activistas, competidores, ambientalistas, medios de comunicación

Un proceso de desarrollo debe servir de guía para todos los participantes y debe poder adaptarse a cada caso particular. Esta adaptación, tal como explica Jacobson (Jacobson I. B., 2000) se produce en el contexto del proyecto, donde se analizan las características del entorno del problema que se debe resolver, de las herramientas con las que se cuenta y, muy especialmente, de las personas que están involucradas en la creación del producto de software. Es importante destacar, que un proceso es una definición de un conjunto de actividades, no su ejecución. La ejecución ocurre en el marco de los proyectos. (Jacobson I. B., 2000).

Existen diversos y diferentes procesos de desarrollo de software, pero todos ellos deben incluir cuatro actividades fundamentales que son comunes, (Sommerville, 2011):

- *Especificación del Software*. Se engloban todas las actividades orientadas a definir la funcionalidad y las restricciones del software para su operación.
- *Desarrollo del Software*. Involucra todas las actividades dedicadas a la producción del software, según lo especificado en la actividad anterior.
- *Validación del Software*. Esta actividad se centra en asegurar que el sistema proporcione lo que el cliente quiere.
- *Evolución del Software*. El software tiene que evolucionar para satisfacer las necesidades cambiantes del entorno.

En cierta forma, estas actividades forman parte de todos los procesos de software, aunque en la práctica son actividades complejas en sí mismas con subactividades, y es allí donde cada autor define a su manera, la forma de nombrarlas, el nivel de detalle con el que las va a dividir, los artefactos⁴ que se obtendrán de cada una, los roles y las herramientas de trabajo.

La definición del dominio del problema y la determinación de los requerimientos⁵ son algunas de las actividades encuadradas dentro de la Especificación del Software y es justamente aquí donde se concentran los mayores problemas a la hora de construir software.

Freeman (Freeman P. , 1987), explica que el concepto de *ciclo de vida* ha sido adoptado en el mundo del software para indicar las etapas por las que el mismo atraviesa. Como en las ciencias biológicas, de las cuales se adoptó el término, una pieza de software ejecutable pasa por un conjunto de etapas que varían de acuerdo con su naturaleza (y en cierto modo por las técnicas que se usan para dar existencia al producto terminado).

⁴Artefacto: resultado que se obtiene de la ejecución de una actividad del proceso de desarrollo

⁵ Requerimientos: también conocidos en la Industria del Software como Requisitos, son la representación de una necesidad, condición o capacidad que proviene del usuario y que el software debe satisfacer.

Existe coincidencia entre los autores, que el software pasa por las siguientes etapas durante su ciclo de vida:

- Desarrollo del concepto (definición)
- Creación técnica (desarrollo)
- Producto para uso (operación)
- Objeto de desarrollo adicional (evolución)

Por su parte McConnell (McConnell S. , 1996) plantea que cada esfuerzo de desarrollo de software atraviesa por un *ciclo de vida*, el cual consiste de todas las actividades entre el tiempo que la primera versión del software comienza como un destello en los ojos de alguien, hasta que la última versión de ese software toma su último aliento en la última máquina de un cliente.

Según lo plantea McConnell, la función principal de los ciclos de vida es establecer el orden en el que un proyecto específico, crea prototipos, diseña, implementa, revisa y ejecuta sus actividades. Establece el criterio para determinar si se continúa de una actividad a la siguiente.

Cuando el proceso implica la construcción de algún producto, se suele referir al proceso como su ciclo de vida. Es por este motivo, que al proceso de desarrollo de software suele denominarse comúnmente ciclo de vida del software, debido a que, en el caso del software, describe la vida de un producto desde su concepción hasta su implementación, entrega, utilización y mantenimiento.

Modelos en el Software y Modelos del Software

En los párrafos anteriores se habló de procesos de desarrollo, a los que se los definió como “*un conjunto organizado de actividades que debía alcanzar un objetivo, cada una de esas actividades genera o produce una salida*”. Jacobson (Jacobson I. B., 2000) define genéricamente como “*artefacto*” a cada resultado que se obtiene de la ejecución de una actividad del proceso de desarrollo. Alguno de los artefactos que se generan son modelos. La construcción de un sistema es por lo tanto un proceso de construcción de modelos, utilizando modelos para representar conocimiento, creados desde perspectivas diferentes y para interesados diferentes.

La importancia de modelar en el software

El modelado es una técnica de ingeniería probada y bien aceptada. El modelado es parte de cualquier disciplina de ingeniería, es parte de cualquier industria. En esta sección se hará énfasis en el modelado como elemento fundamental para el desarrollo de software.

¿Qué es un modelo? Booch, (Booch G. R., 2005), lo define de forma sencilla como “*simplificación de la realidad*”. Un modelo proporciona planos del software, algunos detallados otros más generales. Un buen modelo incluye elementos que se consideran como relevantes para un propósito o para un interesado en particular, y omite aquellos que son menores o no son relevantes.

Ahora bien, esa simplificación de la realidad de la que habla Booch está estrechamente vinculada con un proceso mental que es requerido para construir software: la *abstracción*⁶ De acuerdo con Evans, (Evans, 2004) la abstracción es un modelo del dominio.

Hay una razón fundamental por la cual modelar: “*Construimos modelos para comprender mejor el software que estamos desarrollando*”. (Booch G. R., 2005)

Booch, sintetiza en cuatro a los objetivos pretendidos con el uso de modelos:

1. Ayudan a visualizar cómo es, o cómo queremos que sea un software.
2. Permiten especificar la estructura y/o comportamiento del software.
3. Proporcionan plantillas que nos guían en la construcción de un software.
4. Documentan las decisiones que hemos adoptado.

Hay límites a la capacidad humana de comprender la complejidad. A través del modelado, se reduce el problema que se está estudiando, centrándose sólo en un aspecto a la vez. Es, este esencialmente el enfoque de Dijkstra de “divide y vencerás”.

El uso de modelos ha acompañado la evolución de la disciplina informática desde sus principios, es por ello por lo que se sugiere tener en cuenta, al momento de modelar, algunos principios básicos expuestos por Booch, (Booch G. R., 2005).

El primero de los principios plantea la importancia que tiene la elección de qué modelos crear y la profunda influencia que esto tiene sobre cómo se acomete un problema y cómo se da forma a una solución.

El segundo de los principios define que todo modelo puede ser expresado a diferentes niveles de precisión.

El tercero define que los mejores modelos están ligados a la realidad y por último el cuarto plantea que un único modelo no es suficiente.

Cualquier sistema no trivial se aborda mejor a través de un pequeño conjunto de modelos casi independientes. Esta última expresión “casi independiente”, en este contexto, significa que los modelos que se pueden construir y estudiar en forma separada están interrelacionados. Se puede ejemplificar esta idea, asimilándola con la construcción de un edificio, donde se puede estudiar los planos eléctricos en forma aislada, pero también se debe considerar su correspondencia con los planos de planta e incluso su interacción con el plano de plomería. Lo mismo se puede aplicar para el caso del software. Para comprender

⁶ Abstraer es separar por medio de una operación intelectual las cualidades de un objeto para considerarlos aisladamente o para considerar el mismo objeto en su pura esencia o noción. Abstrarer es captar con el entendimiento el significado o esencia de la cosas. (Castañeda et al., 2007, p. 66).

el software es necesario modelarlo desde varias perspectivas diferentes, comúnmente denominadas “vistas”, que se construyen para diferentes interesados, con niveles de detalle diferentes.

Notaciones y Lenguajes para modelar software

Tal como se menciona en la sección anterior el uso de modelos en la construcción de software tiene tanta historia como el software y ha ido evolucionando en forma paralela a la evolución de las demandas de software, de las tecnologías que se usan para construirlo, con intención de contribuir a la obtención de software de calidad.

Los lenguajes de modelado basados en el paradigma de objetos⁷ comenzaron a utilizarse alrededor de los años ochenta, cuando los metodologistas, de cara a los lenguajes de programación orientados a objetos, y a aplicaciones cada vez más complejas, comenzaron a experimentar con enfoques alternativos de análisis y diseño. Desde ese momento surgieron numerosos métodos orientados a objetos (más de 50 entre 1989 y 1994), (Booch G. R., 2005). Cada uno de ellos, tiene sus propias características, su alcance y notación propios.

Del conjunto de esos métodos, surgieron tres que se destacaron: el método de Booch (Booch G. , 1991), el de Jacobson OOSE (Jacobson I. , 1994) y el de Rumbaugh (Rumbaugh, 1996) y que luego a fines de 1994 decidieron unirse para crear un lenguaje de modelado que sea unificado, que permita modelar sistemas desde el concepto hasta los artefactos ejecutables, utilizando el paradigma de orientación a objetos como marco de referencia para la representación de la realidad. Este lenguaje debía servir para especificar, documentar y modelar cualquier tipo de sistema de software, que sea interpretado tanto por personas como por computadoras.

Este lenguaje conocido en la actualidad como UML (Unified Modeling Lenguaje) en su versión 1.0 fue liberado y puesto a disposición como estándar en la OMG⁸ (Object Management Group) en enero de 1997. En el año 2005, aparece la versión vigente que es el UML 2.0, que también fue aprobado, ese mismo año, como estándar por la ISO⁹ como ISO/IEC 19501:2005 Information Technology. El UML fue pensado fundamentalmente para modelar software, sin embargo, puede utilizarse también en otros ámbitos como ser el modelado de procesos de negocio.

Booch, (Booch G. R., 2005) describe a los diagramas como un representación gráfica de un conjunto de elementos, visualizado la mayoría de las veces como grafo conexo de nodos (elementos) y arcos

⁷Paradigma de Objetos: un paradigma representa un enfoque particular o filosofía para construir software que conceptualiza la manera que se tiene de visualizar a la realidad. En el caso del paradigma de objetos, este propone la representación de la realidad como un conjunto de objetos, que colaboran entre ellos para realizar tareas. Los objetos son entidades que tienen un determinado estado (valor que asumen sus atributos, comportamiento (conjunto de operaciones que puede realizar) e identidad (lo que permite reconocerlo y lo distingue de los demás objetos).

⁸OMG: Es un consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA. Es una organización sin ánimo de lucro que promueve el uso de tecnología orientada a objetos mediante guías y especificaciones para las mismas. El grupo está formado por diversas compañías y organizaciones con distintos privilegios dentro de la misma. (http://es.wikipedia.org/wiki/Object_Management_Group).

⁹ ISO: International Organization for Standardization.

(relaciones). Los diagramas se dibujan para visualizar un sistema desde diferentes perspectivas, de forma que un diagrama es una proyección de un sistema. Un diagrama representa una vista resumida de los elementos que constituyen un sistema.

El valor agregado fundamental que el UML aporta es brindar la posibilidad de “comunicación” entre todas las personas involucradas en la construcción de un software, que trasciende el espacio educativo donde se formarán, el ámbito donde trabajan, la tecnología o el proceso elegido para construirlo.

Esto cambia el vínculo entre los involucrados desde la concepción y el modelado del sistema, hasta su utilización final por parte de los usuarios; modificando las relaciones entre las personas, lo cual podemos considerarlo como el aporte más importante del UML, es decir el cambio en las relaciones de los *stakeholders*. Este aspecto es tributario a la idea del presente trabajo.

Procesos para el Desarrollo de Software

El Software Engineering Institute (SEI) (Software Engineering Institute, 2010), en sus investigaciones para ayudar a las organizaciones a desarrollar y mantener productos y servicios de calidad, ha identificado varias dimensiones en las que una organización puede centrarse para mejorar su actividad. La Figura 3 ilustra las tres dimensiones críticas donde normalmente se centran las organizaciones: *las personas, los métodos y procedimientos*, el equipamiento y herramientas. Lo que mantiene todo unido son los procesos utilizados en la organización. Éstos le permiten alinear su forma de trabajo. Permiten abordar la escalabilidad y proporcionan un método de incorporar conocimiento de cómo hacer mejor las cosas. Los procesos les permiten a las organizaciones utilizar mejor sus recursos, evaluar y analizar las tendencias de su actividad.

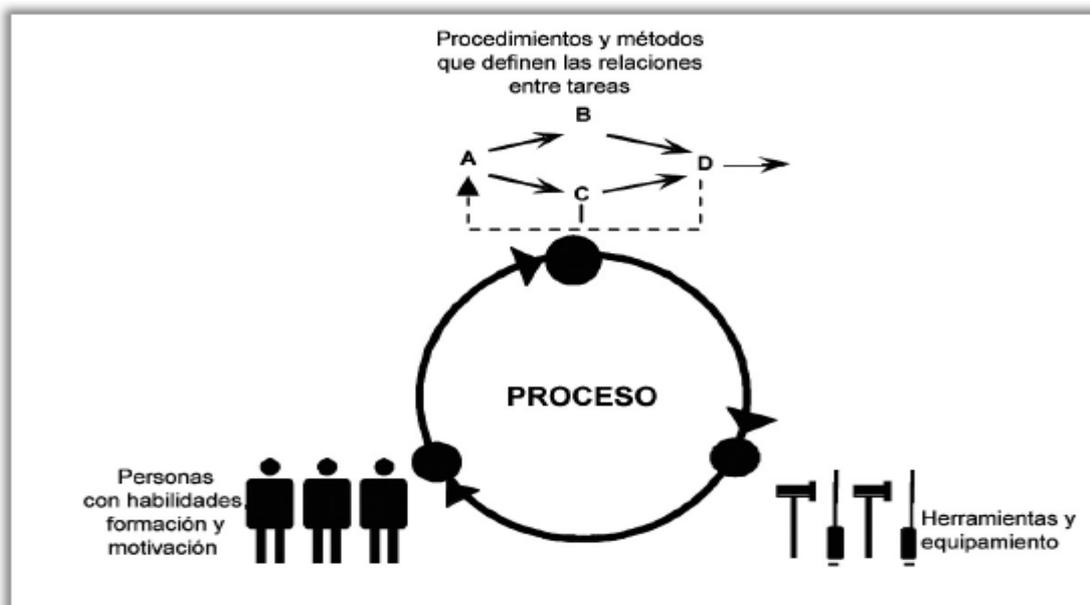


Figura 3. Las tres dimensiones críticas
Fuente (Software Engineering Institute, 2010)

La industria del software ha reconocido la importancia de trabajar con procesos, estos les permiten a las organizaciones lograr los objetivos de negocio de una forma racionalizada. Estos procesos permiten alcanzar los objetivos de negocio de las organizaciones.

En la actualidad cohabitan en la industria, dos tipos de procesos: procesos definidos y procesos empíricos.

Procesos definidos – Proceso Unificado de Desarrollo

Los procesos definidos son lo que Deming, citado en (Humphrey, 1995) llama *definiciones operacionales*: algo que cada persona pueda comunicar y con lo que pueda trabajar. Estas definiciones operacionales según Kellner citado en (Humphrey, 1995), proveen los siguientes beneficios:

- Permiten una comunicación efectiva de los procesos a los involucrados relevantes: usuarios, desarrolladores, administradores, clientes e investigadores.
- Mejoran la comprensión de la administración, proveen una base precisa para la automatización del proceso y facilitan la movilidad del personal.
- Facilitan el reuso del proceso, dado que el proceso de desarrollo consume tiempo y es caro.
- Soportan la evolución del proceso brindando un medio efectivo para el aprendizaje del proceso y una base sólida para su mejora.
- Ayudan a la administración del proceso puesto que la administración requiere planes claros y precisos contra los cuales medir el estado y los procesos definidos les dan ese contexto.

Los procesos definidos o predictivos tienen las siguientes características:

- Asumen que es posible predecir y detallar a largo plazo las variables del proyecto. Se intenta solucionar la complejidad de un proyecto esforzándose en una planificación detallada. Para no desviarse de la predicción inicial, este proceso necesita de un férreo control de cambios de requisitos.
- La comunicación con el cliente sobre el resultado a entregar se basa en la elaboración de un documento que incluye todo el alcance del proyecto. Documento que se espera que el cliente valide. Esta actividad debe realizarse de manera temprana, al inicio del proyecto.
- El proyecto está preparado para ser entregado hacia su final (o al final de una fase de varios meses), momento en que existe el riesgo de la insatisfacción del cliente por recibir un producto que no es el esperado.

Estos procesos se han basado históricamente, en torno a las habilidades de individuos altamente cualificados que saben cómo hacer su trabajo y lo hacen bien. Sin embargo, no cabe duda de que estos desarrolladores necesitan dirección organizativa, las que llamamos "*proceso de desarrollo de software*", estos procesos de desarrollo han sido llevados a cabo por diferentes metodologías para el desarrollo de software. A fines de la década del noventa estas metodologías se unifican en lo que se da en llamar *el*

Proceso Unificado de Desarrollo (Jacobson I. B., 2000). En este proceso se define *quién* está haciendo *qué*, *cuándo* y *cómo* alcanzar un determinado objetivo; considerando en términos de la ingeniería de software que el objetivo es construir un producto software o mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad, encapsulando las mejores prácticas que el estado actual de la tecnología facilita.

Es así como el proceso del software se reduce a la dificultad que afrontan los desarrolladores para coordinar las múltiples cadenas de trabajo de un proyecto de software.

Los métodos definidos, proporcionan una guía para ordenar las actividades de un equipo, dirigen las tareas de cada desarrollador por separado y del equipo como un todo, especifican los artefactos que deben desarrollarse y ofrecen criterios para el control y medición de los productos y actividades de un proyecto.

Específicamente el proceso unificado (PUD) es un proceso de desarrollo de software, que transforma los requisitos de un usuario en un sistema de software, (Jacobson I. B., 2000); es un marco de trabajo genérico que puede especificarse para una gran variedad de sistemas software, diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de actitud y diferentes tamaños del proyecto, como se ve representado en la Figura 4.

El PUD está centrado en la arquitectura, considerando a ésta como diferentes vistas del sistema en construcción. El PUD es iterativo e incremental, considera que es práctico dividir el trabajo en partes más pequeñas o mini proyectos, repitiendo a lo largo de una serie de ciclos que constituyen la vida de un sistema; cada uno de esos ciclos concluye en una versión del producto que es presentada a los usuarios. Por último, el PUD es dirigido por casos de uso¹⁰, lo que significa que el proceso de desarrollo sigue un hilo conductor, avanza a través de una serie de flujos de trabajo que parten de cada caso de uso, éstos se especifican, se diseñan y son fuente para la construcción de las pruebas del sistema, es decir los casos de uso guían la arquitectura del sistema y la arquitectura del sistema influye en la selección de los casos de uso.

Los procesos definidos y más específicamente el PUD consideran que un proceso de desarrollo de software guía los esfuerzos de las personas implicadas en el proyecto, explicando los pasos necesarios para la ejecución de este. El proceso está automatizado por herramientas como se ve en la Figura 4

¹⁰ **Caso de uso:** un caso de uso representa una forma de utilizar el software, descrito desde la perspectiva del usuario final y focalizando en la interacción entre el usuario y el sistema.

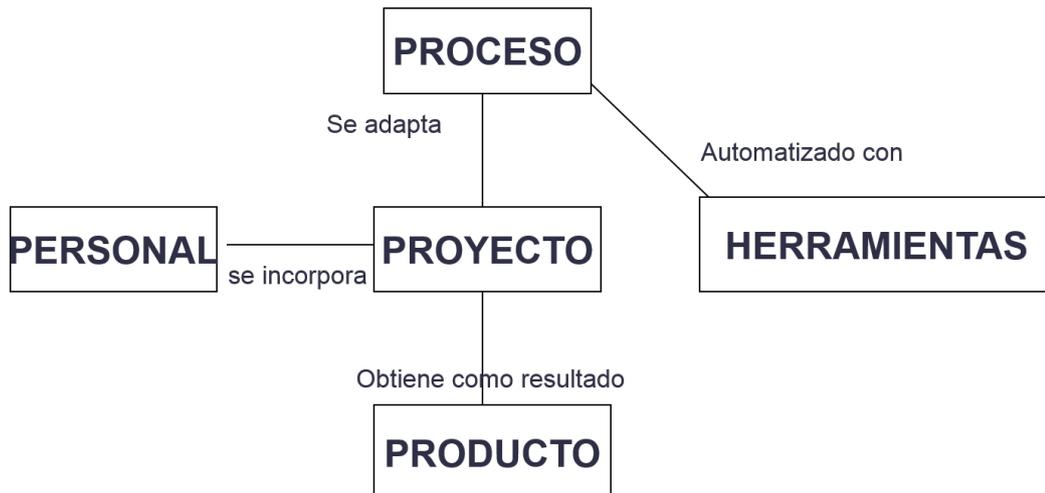


Figura 4. Las 4 “P” en el desarrollo de software
Adaptada de (Jacobson I. B., 2000)

Las personas son los principales autores de un proyecto software del cual participan, arquitectos, desarrolladores, testadores, administradores, usuarios, clientes y otros interesados. El proyecto es el elemento organizativo a través del cual se hace la gestión del desarrollo de software en tanto, el producto es el artefacto que se crea a lo largo de la vida del proyecto. El proceso es el conjunto de actividades necesarias para transformar los requisitos de los usuarios en un producto y por último las herramientas son el software que se utiliza para crear más software durante el conjunto de actividades definidas en el proceso.

Las personas son decisivas

Hay personas implicadas en el desarrollo de un producto software durante todo su ciclo de vida. Financian el producto, lo planifican, lo desarrollan, lo gestionan, lo prueban, lo utilizan. Por lo tanto, el proceso que guía este desarrollo debe orientarse a las personas. Es decir, debe funcionar bien para las personas que lo utilizan.

Procesos empíricos- Métodos Ágiles

Los procesos empíricos son aquellos que parten de un conjunto de definiciones iniciales respecto del producto y del proyecto, que luego se van adaptando en ciclos cortos de tiempo, a las necesidades de los interesados. Estos procesos tienen las siguientes características:

- Asumen que hay un horizonte de predicción de las variables del proyecto dado que siempre hay cambios en el contexto del proyecto, debido a la indeterminación y complejidad propios.
- Para gestionar la complejidad y obtener el mayor valor posible, el proceso se base en inspección y adaptación regular en función de los resultados parciales que se van obteniendo y del propio contexto del proyecto.

Tomando los principios de los procesos empíricos antes definidos, surgen las llamadas metodologías ágiles.

Tal como se mencionó en secciones anteriores, las metodologías tradicionales imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Lo hacen desarrollando un proceso detallado con un fuerte énfasis en planificar, inspirado en otras disciplinas de la ingeniería. La crítica más frecuente a estas metodologías es que son burocráticas y “pesadas”. (Fowler, 2005).

La familia de métodos de desarrollo ágiles evolucionó a partir de los conocidos ciclos de vida iterativos e incrementales. Surgieron de la creencia que un acercamiento más en contacto con la realidad social y la realidad del desarrollo de productos basados en el aprendizaje, la innovación y el cambio daría mejores resultados.

El propósito principal de estas metodologías es lograr un equilibrio entre los procesos disciplinados que proponen una excesiva carga de trabajo y burocracia, y la no existencia de proceso, que lleva a desarrollar software de manera caótica.

La alternativa que ofrecen las metodologías ágiles propone un cambio radical que hace énfasis en un replanteo de aspectos culturales, comunicacionales y sociales, que son los principales causantes de fracasos en los proyectos de desarrollo de software.

El manifiesto ágil (Agile Alliance, 2001), refleja el acuerdo de la mayoría de los referentes de la industria de software que plantean prácticas ágiles. Este manifiesto, en su declaración contiene valores y principios a los que quienes adhieren a él, se comprometen a respetar.

Este manifiesto gira en torno de privilegiar lo que es de valor para el cliente, es decir, entregas frecuentes de software funcionando. Propone un ambiente de colaboración con el cliente/usuario y entre los miembros del equipo y una actitud de aceptación frente a los cambios que surjan durante el desarrollo del producto de software. En este sentido se pone énfasis, en dar más importancia a las personas involucradas en el desarrollo de software que a las herramientas y procesos que se utilizarán para construirlo.

También se destaca la incidencia que tiene la presencia y colaboración del cliente para alcanzar el éxito del proyecto.

La propuesta más desafiante del manifiesto ágil tiene que ver con evaluar cada acción, cada decisión, cada elección con la vara de entregar “valor de negocio” a quién solicita el software, que a su vez se siente parte del proyecto.

Existen actualmente un gran número de métodos ágiles, que suscriben al manifiesto. Algunos de los cuales definen prácticas para desarrollo ágil, otros para modelado ágil y otros para gestión de proyectos

ágil. Los métodos más utilizados en la actualidad son SCRUM (Schwaber K. &, 2001) para gestión de proyecto y XP (Beck, 2004) (Extreme Programming) para ingeniería. Ambos métodos se describen sucintamente en las próximas secciones.

SCRUM

Scrum (Schwaber K. &, 2013) es un entorno de trabajo que define prácticas para la gestión de proyectos. Si bien puede utilizarse para proyectos de propósitos diversos, se hará énfasis en proyectos de desarrollo de software.

Scrum propone un ciclo de vida iterativa e incremental, que utiliza para liberar versiones frecuentes del software, acotadas en funcionalidad, al final de cada iteración a la que llama "Sprint". Cada sprint es un período fijo de tiempo, con una duración que oscila entre dos y cuatro semanas, encuadrado en la característica de "timeboxing", lo que significa que la duración en tiempo es fija.

Si bien la propuesta es que al final de cada sprint se obtenga un producto potencialmente entregable, o sea software en condiciones de ser instalado y funcionar, muchas organizaciones eligen no liberar nueva funcionalidad del producto al fin de cada sprint. En lugar de eso, prefieren combinar el resultado de varios Sprints en un único entregable a liberar (release).

A continuación, se describe esquemáticamente la propuesta del framework Scrum, (Figura 5). El framework está compuesto por tres roles, tres entregables y cuatro reuniones o ceremonias (Schwaber K. &., 2013).

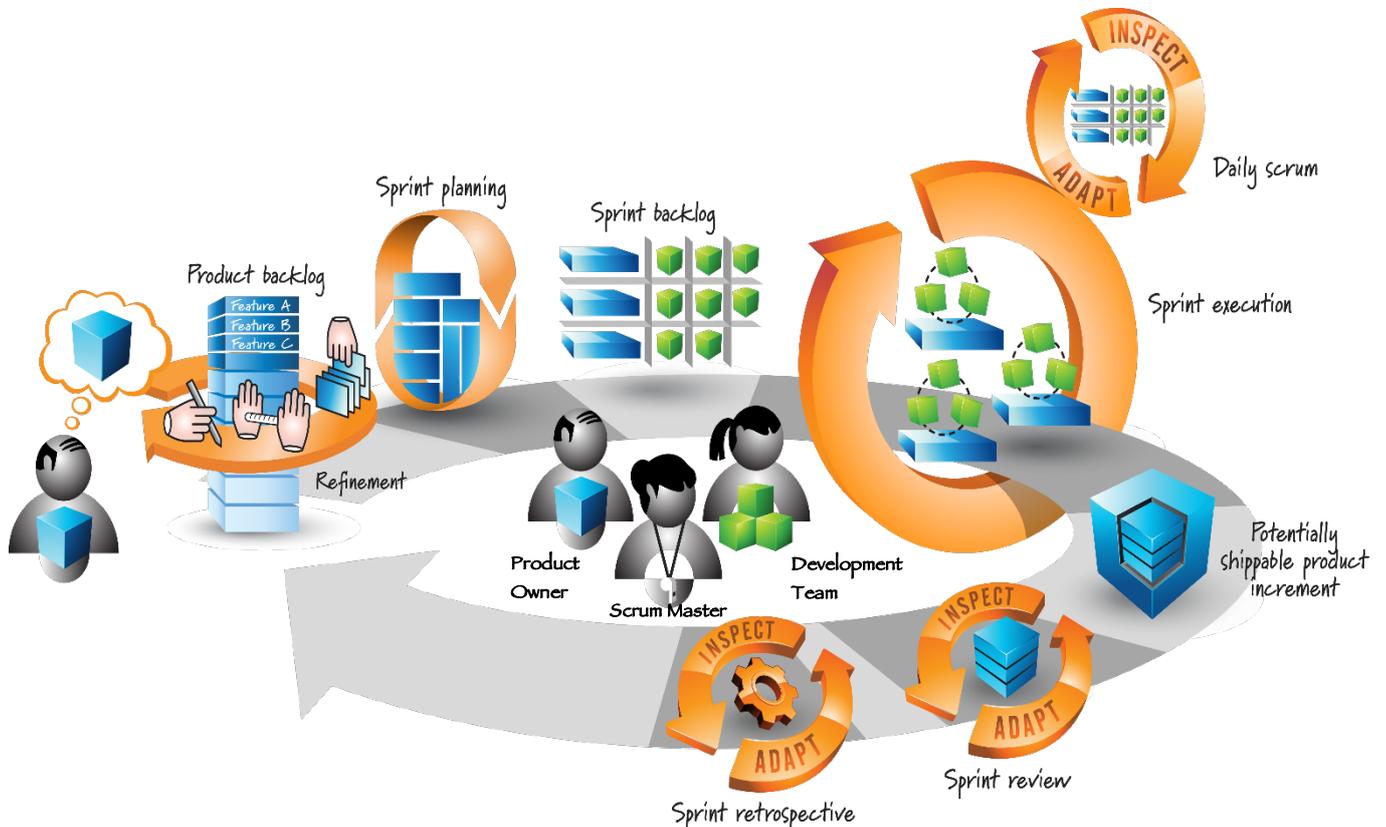


Figura 5. Ciclo de Vida de Scrum
Adaptado de (Rubin, 2013)

Los roles definidos, que constituyen el Equipo Scrum son:

- *Scrum Master*: es el líder de servicio, cuya responsabilidad más importantes es velar por el cumplimiento de las reglas de Scrum, como así también, dar soporte al resto del equipo y actuar como facilitador neutral preocupado en fomentar un ambiente colaborativo y en guiar al equipo hacia el mejoramiento continuo y la autosuficiencia.
- *Product Owner*: el dueño del producto, la voz principal del cliente, quien asume la responsabilidad por su construcción y por maximizar el valor de negocio.
- *Development Team*: el equipo de desarrollo es el responsable de construir el producto definido por el Product Owner. Cada miembro del equipo en Scrum es “multifuncional” – tiene todas las competencias y habilidades necesarias para liberar un producto potencialmente entregable al final de cada iteración. Es un equipo “autoorganizado” (autogestionado), con un alto grado de autonomía y responsabilidad. En Scrum, los equipos se autoorganizan en vez de ser dirigidos por un Líder de Proyecto.

Propone la generación de tres entregables:

- *Product Backlog*: es la lista de características que se espera tenga el producto de software. Estas características están priorizadas y pueden expresarse a distintos niveles de granularidad. Es importante destacar que es una lista dinámica que está en constante evolución y refinamiento.
- *Sprint Backlog*: Lista de elementos, tomada de la parte superior del product Backlog, que contiene la porción de características que el equipo acordó que puede construir durante una iteración (Sprint). Estas características deben estar priorizadas por el Product Owner y estimadas por el equipo.
- *Incremento del Producto*: versión del software, potencialmente desplegable en el ambiente de producción, obtenida como resultado de la ejecución de un sprint.

A su vez propone cuatro tipos de eventos o ceremonias:

- *Sprint Planning Meeting*; reunión de planificación del sprint: es la primera ceremonia, cuyo propósito es obtener la definición del incremento del producto que se liberará en ese sprint. En la primera parte de la reunión, la reunión del qué, se define el objetivo del sprint y el Sprint Backlog. En la segunda parte de la reunión, la reunión consiste en abordar el cómo, el equipo estima las características y puede descomponerlas en las tareas que necesitará hacer para desarrollarlas.
- *Daily Meeting*: en esta reunión se une al equipo, de pie en lo posible. Todos deben asistir, la duración aproximada recomendada es de 15 minutos. El propósito fundamental es dar visibilidad y sincronizar diariamente al equipo. Cada integrante a su turno debe responder tres preguntas: ¿Qué hizo desde la reunión anterior? ¿Qué piensa hacer hasta la próxima reunión? ¿Qué impedimentos se le presentaron?
- *Sprint Review Meeting*: la reunión de revisión, también conocida como “Sprint Demo”, es el momento en el cual el equipo le presenta al Product Owner el incremento del producto que obtuvo durante la ejecución del sprint. Se hace al final del sprint, su propósito es obtener realimentación sobre el producto. El Product Owner evalúa las funcionalidades desarrolladas y determinará cuáles va a aceptar y cuáles no.
- *Sprint Retrospective Meeting*: Está es la última reunión que se realiza en un Sprint, el propósito es obtener realimentación del proceso utilizado, repasar aspectos positivos y negativos y definir acciones a seguir para mejorar la calidad del proceso.

Si bien Scrum propone 4 ceremonias, hay una quinta ceremonia que, a pesar de no estar explícitamente definida en el framework, formalizarla es importante y necesario. En este evento denominado Product Backlog Refinement (Scrum Alliance), el Product Owner tiene un rol fundamental, y es la reunión de creación, preparación y mantenimiento del Product Backlog. Esta reunión no tiene un momento fijo predefinido de realización, no obstante, se aconseja realizarla en las siguientes oportunidades: a) para definir el Product Backlog por primera vez, b) antes de cada Sprint Planning y c) en cualquier momento que el equipo de desarrollo considere necesario, durante la ejecución de un Sprint, para comprender y refinar los requisitos que se están desarrollando.

Scrum propone un nivel bajo de ceremonia en el proceso, lo que significa que son pocas prácticas las que define, no obstante, no se debe confundir ágil con débil o permisivo, dado que el método es estricto en el cumplimiento de sus definiciones y sus reglas.

Por ser Scrum un framework de “gestión”, no define lineamiento alguno respecto a cuestiones vinculadas a la ingeniería del producto, ni da detalle sobre cómo construir el software, asumiendo que el equipo de desarrollo sabe, está capacitado y en condiciones, para resolver esa misión.

Las diferencias que pueden destacarse de este enfoque de gestión ágil respecto de la gestión tradicional de proyectos están sustentadas fuertemente en el compromiso, la autoorganización, la autogestión y el esfuerzo colaborativo. Son estas las competencias, y aptitudes que deben desarrollarse en las personas para obtener los resultados esperados.

XP (Programación Extrema)

Las raíces de la XP yacen en la comunidad de Smalltalk, y en particular la colaboración cercana de Kent Beck y Ward Cunningham a finales de los años ochenta. Ambos refinaron sus prácticas en numerosos proyectos a principios de los noventa, extendiendo sus ideas de un desarrollo de software adaptable y orientado a la gente. (Fowler, 2005)

La XP empieza con cuatro valores: Comunicación, Retroalimentación, Simplicidad y Coraje. Construye sobre ellos una docena de prácticas que los proyectos XP deben seguir. Muchas de estas prácticas son técnicas antiguas, tratadas y probadas, aunque a menudo olvidadas por muchos, incluyendo la mayoría de los procesos planeados. Además de resucitar estas técnicas, la XP las combina en un todo sinérgico dónde cada una refuerza a las demás, tal como lo muestra la figura 3.5. (Fowler, 2005).

El ciclo de vida ideal de XP consiste de seis fases (Beck, 2004): Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto, según muestra la Figura 6 el ciclo de vida propuesto por XP.

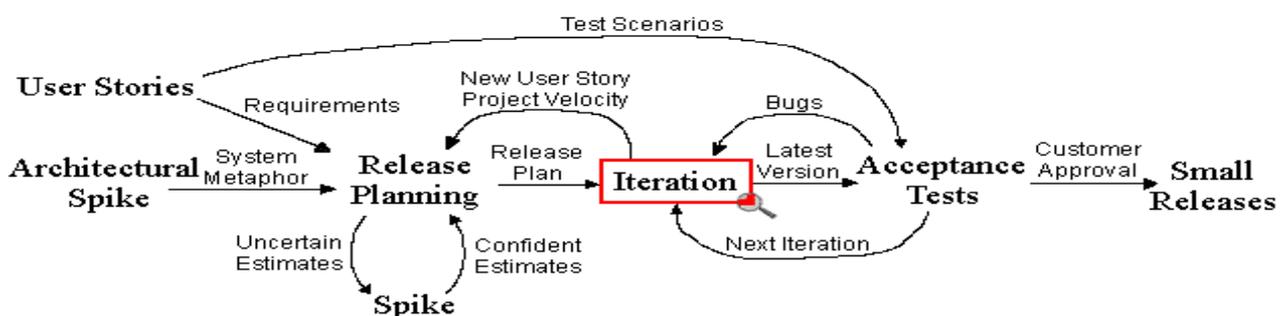


Figura 6. Ciclo de vida de XP
Fuente J. Doesvan Wells

La principal suposición que se realiza en XP es la posibilidad de disminuir la mítica curva exponencial del costo del cambio a lo largo del proyecto. Esto se consigue gracias a las tecnologías disponibles para

ayudar en el desarrollo de software y a la aplicación disciplinada de las prácticas (Letelier & Penadés, 2006), que se describen a continuación.

- *El juego de la planificación*: Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración. El cliente establece la prioridad de cada historia de usuario¹¹, de acuerdo con el valor que aporta para el negocio. Los programadores estiman el esfuerzo asociado a cada historia de usuario. Se ordenan las historias de usuario según prioridad y esfuerzo, y se define el contenido de la entrega y/o iteración, apostando por enfrentar lo de más valor y riesgo cuanto antes.
- *Entregas pequeñas*: La idea es producir rápidamente versiones del sistema que sean operativas, y que constituyan un resultado de valor para el negocio.
- *Metáfora*: En XP no se enfatiza la definición temprana de una arquitectura estable para el sistema, dado que se asume que dicha arquitectura es evolutiva. El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema, según explica Fowler en (Letelier & Penadés, 2006). La práctica de la metáfora consiste en formar un conjunto de nombres que actúen como vocabulario para hablar sobre el dominio del problema. Este conjunto de nombres ayuda a la nomenclatura de clases y métodos del sistema.
- *Diseño simple*: Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente.
- *Pruebas*: La producción de código está dirigida por las pruebas unitarias, conocido como TDD, Test Driven Development (Beck, 2004). Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse.
- *Refactorización (Refactoring)* La refactorización es una actividad constante de reestructuración del código con el objetivo de remover duplicación de código, mejorar su legibilidad, simplificarlo y hacerlo más flexible para facilitar los posteriores cambios.
- *Programación en parejas*: Toda la producción de código debe realizarse con trabajo en parejas de programadores. Las principales ventajas de introducir este estilo de programación son (Henrik,

¹¹ Historia de usuario: Descripción corta de una funcionalidad valuada por un usuario o cliente de un sistema.

2007), mejora en la calidad del código que se produce; en la concentración del equipo y en la distribución del conocimiento entre el equipo.

- *Propiedad colectiva del código:* Cualquier programador puede cambiar cualquier parte del código en cualquier momento. Esta práctica motiva a todos a contribuir con nuevas ideas en todos los segmentos del sistema, evitando a la vez que algún programador sea imprescindible para realizar cambios en alguna porción de código.
- *Integración continua:* Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día. Todas las pruebas son ejecutadas y tienen que ser aprobadas para que el nuevo código sea incorporado definitivamente. La integración continua a menudo reduce la fragmentación de los esfuerzos de los desarrolladores por falta de comunicación sobre lo que puede ser reutilizado o compartido.
- *40 horas por semana:* Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. El trabajo extra desmotiva al equipo.
- *Cliente in-situ:* El cliente tiene que estar presente y disponible todo el tiempo para el equipo. Gran parte del éxito del proyecto XP se debe a que es el cliente quien conduce constantemente el trabajo hacia lo que aportará mayor valor de negocio y los programadores pueden resolver de manera inmediata cualquier duda asociada. La comunicación oral es más efectiva que la escrita, ya que esta última toma mucho tiempo en generarse y puede tener más riesgo de ser mal interpretada.
- *Estándares de programación:* XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios.

La mayoría de las prácticas propuestas por XP no son novedosas sino que en alguna forma ya habían sido propuestas en la ingeniería del software e incluso demostrado su valor en la práctica. El mérito de XP es integrarlas de una forma efectiva y complementarlas con otras ideas desde la perspectiva del negocio, los valores humanos y el trabajo en equipo, como se muestra en la Figura 7.

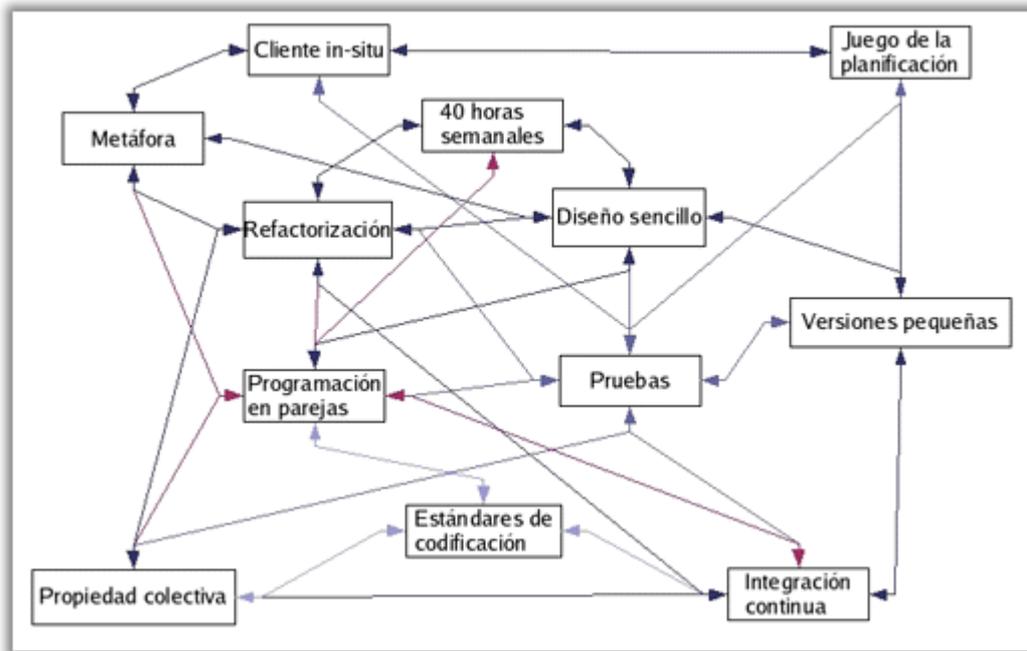


Figura 7. Relación entre las prácticas de XP
Fuente (Letelier & Penadés, 2006)

El Proceso Grupal

Dado el proceso software, entre los aspectos fundamentales del mismo se deben considerar los atributos comunicacionales. De cara a una mejor comprensión de estos fenómenos relacionados con la comunicación se tendrán en cuenta los desarrollos teóricos de una de las disciplinas modernas que aborda esta cuestión, la psicología social que se inscribe en una crítica en la vida cotidiana.

La psicología social aborda al sujeto inmerso en sus relaciones cotidianas y en este caso lo veremos desde la cotidianidad del trabajo, tiene como objeto de estudio el desarrollo y transformación de una relación dialéctica, la que se da entre estructura social y fantasía inconsciente del sujeto asentada sobre sus relaciones de necesidad.

El ser humano es un ser de necesidades, que sólo se satisfacen socialmente en relaciones que lo determinan. El sujeto no es solo un sujeto relacionado, es un sujeto producido en una praxis. No hay nada en el que no sea la resultante de la interacción entre individuos, grupos y clases. La psicología social como disciplina indaga la interacción en sus dos aspectos, ínter subjetivo (grupo externo) e intra subjetivo (grupo interno), es significativa, direccional y operativa. Su punto de partida es una práctica.

Para los grupos de desarrollo de software las relaciones presentes en los procesos de trabajo y la trama vincular en la praxis del proceso software, resulta un eje sustantivo en nuestro análisis, dado que el texto precedente, describe con claridad, los comportamientos de los grupos de desarrollo de software, por la heterogeneidad de este, por las exigencias a las cuales este grupo es sometido y por la complejidad del producto que desarrollan.

Un elemento conceptual que la psicología social provee y resulta de utilidad para pensar el proceso de desarrollo del software es el esquema *conceptual referencial y operativo* (ECRO)

El ECRO es una técnica de los grupos operativos, desarrollada por Pichón Rivière (Riviere, 1997) en la década del '40. El grupo debe configurar un ECRO donde las contradicciones del campo de trabajo deban ser resueltas en la misma tarea grupal.

Para Pichón Rivière el ECRO es un conjunto organizado de conceptos generales, teóricos, referidos a un sector de lo real, a un determinado universo de discurso, que permitan una aproximación instrumental al objeto particular (concreto). El método dialéctico fundamenta este ECRO y su particular dialéctica (Lema, 1986)

Cuando decimos "esquema" hablamos de un conjunto articulado de conocimientos, y entendemos por "esquema conceptual" a los sistemas de ideas que alcanzan una vasta generalización. Son síntesis, más o menos generales, de proposiciones que establecen las condiciones según las cuales se relacionan entre sí los fenómenos empíricos. Es un conjunto de conocimientos que proporciona líneas de trabajo e investigación. Pero la investigación psicológica, o cualquier tipo de tarea científica, sin un adecuado sistema conceptual sería ciega e infructuosa. El descubrimiento se hace posible por la adecuación del esquema conceptual del investigador a las características del fenómeno a investigar. Se trata entonces de una adecuación a las hipótesis. En síntesis, un esquema conceptual es un conjunto organizado de conceptos universales que permiten una aproximación adecuada al objeto particular. Una vez elaborado el esquema conceptual, su transmisión es rápida, fácil, es decir, el esquema conceptual es aprehensible y transmisible. El ECRO resulta un modelo en virtud de su carácter de instrumento para la aprehensión de la realidad, es por lo tanto un instrumento de comprensión del sector de la realidad que se propone estudiar. El ECRO permite una comprensión horizontal (las relaciones sociales, la organización y el sistema social) y vertical (el individuo inserto en este sistema) de una sociedad en permanente situación de cambio y los problemas de adaptación o de relación del individuo con su medio.

Es posible, a partir de lo dicho, definir al ECRO como un *"conjunto organizado de nociones y conceptos generales teóricos, referidos a un sector de lo real, a un universo del discurso, que permite una aproximación instrumental al objeto particular concreto. Este ECRO y la didáctica que lo vehiculiza están fundados en el método dialéctico"*. (Pichon-Rivière, 2002)

El esquema se caracteriza como conjunto organizado de nociones y conceptos generales, teóricos, referidos a un sector de lo real, a un universo del discurso, que permite una aproximación instrumental al objeto particular concreto.

Todo esquema conceptual, referencial y operativo tiene un aspecto superestructural y otro aspecto infraestructural. Lo superestructural está dado por los elementos conceptuales y lo infraestructural por los elementos emocionales, motivacionales, es decir lo que nosotros denominaríamos verticalidad del sujeto,

elementos éstos surgidos de su propia experiencia de vida y que determinan las modalidades del abordaje de la realidad (Lema, 1986)

El método dialéctico por el que se desarrolla la espiral del conocimiento implica un tipo de análisis que devela los principios opuestos, las tendencias contradictorias, fuentes configuradoras de la dinámica de los procesos. La espiral dialéctica, como método de aprendizaje es el que mejor representa los procesos de aprendizaje de los grupos de desarrollo.

El esquema referencial grupal es un esquema común a los miembros del grupo quienes, al compartir un código común, tienen facilitada la comunicación intragrupal.

Este método tiene un paralelismo notable con los ciclos de vida de desarrollo del software, de aprendizaje permanente e incremental. Observamos que los procesos elicitorios, de encuentro y de trabajo entre todos los stakeholders, son eminentemente dialécticos.

La noción de aprendizaje se vincula íntimamente con el criterio de adaptación activa a la realidad. Aprender es realizar una lectura de la realidad, es decir, una lectura que implique capacidad de evaluación y creatividad (transformación de lo real), es de subrayar que estos conceptos son incorporados por el manifiesto ágil (ya descrito) casi en su totalidad.

El método dialéctico por el que se desarrolla la espiral del conocimiento implica un tipo de análisis que, a partir de los hechos elementales, las relaciones cotidianas, devela los principios opuestos, las tendencias contradictorias, fuentes configuradoras de la dinámica del proceso. Este método es el que permite la producción del conocimiento de las leyes que rigen la naturaleza, la sociedad y el pensamiento, tres aspectos de lo real comprometidos en lo que denominamos "*hombre-en situación*". Con el término "*hombre-en-situación*" se pretende caracterizar un objeto de conocimiento, en una tarea que reintegre lo fragmentado por un pensamiento disociado que escotomiza las relaciones entre sujetos, naturaleza y sociedad.

El desarrollo de un ECRO común a los miembros del grupo permite el incremento de la comunicación intragrupal ya que, de acuerdo con la teoría de la información, lo que permite que el receptor comprenda el mensaje emitido por el transmisor, a través de operaciones de codificación y decodificación, es una semejanza de esquemas referenciales. En este proceso de comunicación y aprendizaje observamos que el grupo sigue un itinerario que va del lenguaje común al lenguaje científico.

Pichón Rivière sostiene que los procesos de codificación y decodificación de señales pertenecen a esquemas referenciales individuales y de los grupos a través de los que se hace posible, según sea el funcionamiento y la estructura de estos esquemas, configurar situaciones de entendimiento o mal entendimiento. En última instancia, la comunicación grupal es posible por la existencia de un ECRO grupal.

El grupo operativo como el instrumento de trabajo:

Se define al grupo como el conjunto restringido de personas ligadas entre sí por constantes de tiempo y espacio y articuladas por su mutua representación interna, que se propone en forma explícita o implícita una tarea de constituye su finalidad.

La técnica operativa está centrada en la tarea. Toda situación de aprendizaje genera en los sujetos dos miedos básicos.

A) Miedo a la pérdida del equilibrio ya logrado en la situación anterior.

B) Miedo al ataque en la nueva situación en la que el sujeto no se siente adecuadamente instrumentado. Ambos miedos que coexisten y cooperan configuran la resistencia al cambio. Resistencia que se expresa en términos de dificultades en la comunicación y el aprendizaje. Itinerario del grupo y relaciones cotidianas:

El grupo operativo es la primera instancia de lo cotidiano. El acontecer del grupo centra a la investigación del aprendiz de la psicología social en el fenómeno universal de la interacción de donde surge el reconocimiento de sí y del otro en un diálogo y un intercambio permanente que sigue una trayectoria en espiral.

En esta dinámica, se observa que el coordinador mantiene con el grupo una relación asimétrica. El equipo de coordinación, integrado por coordinador y observador, cada uno desde su rol específico y a partir de un ECRO que le permite la comprensión de las leyes estructurantes del proceso grupal, detecta las situaciones significativas (emergentes) que desde lo explícito remiten como signo a formas implícitas de interacción.

La interpretación operativa modifica el campo grupal, permite a partir del autoconocimiento la reestructuración de las relaciones entre los miembros y con la tarea. Esa así como la interpretación incluye explícita o implícitamente un criterio de realidad o esquema referencial a partir del cual se analiza la situación del grupo. Por lo tanto se puede concluir que consiste en la decodificación del sentido de lo emergente; y en este orden de ideas, es un aporte de significados al grupo.

Estos aspectos recuperados de la Psicología Social son atributos que nos permiten ver como esta disciplina tiene un paralelismo significativo y puede iluminar nuestros grupos de trabajo de desarrollo de software, nuestros roles y nuestro esquema metodológico de operar.

Tomando el esquema ECRO se propone identificar en el grupo de desarrollo software (que implica en principio constituirse como tal en términos de mutua representación interna y en constantes de espacio-tiempo (en la actualidad con peculiaridades propias de la era digital) características que son sustantivas para comprender las relaciones y vínculos interpersonales, la relación con la tarea (involucramiento y compromiso) papel del líder y conductor del grupo, los miedos y resistencias que pueden surgir en un proceso creativo que constituye el desarrollo (como el aprendizaje espiralado), los desequilibrios, el lugar que ocupa la dialéctica, la adaptación activa a la realidad y la participación de stakeholders durante el proceso de desarrollo.

Respecto a la comunicación, Pichón. Riviere plantea: *"Comunicación es todo proceso de interacción social por medio de símbolos y sistemas de mensajes. Incluye todo proceso en el cual la conducta de un ser humano actúa como estímulo de la conducta de otro ser humano. Puede ser verbal, o no verbal, interindividual o intergrupal"* (Riviere, 1997)

El cono invertido

El cono invertido es un recurso visual para comprender mejor cómo se lleva a cabo el análisis sistemático de las situaciones grupales, entendiendo por análisis hacer explícito lo implícito. El esquema está constituido por tres elementos: la base, el vértice y la espiral dialéctica. El esquema del cono invertido tiene la intención de configurar en su base todas las situaciones manifiestas en el campo operacional y en su vértice, las situaciones básicas universales que están actuando en forma latente. La espiral dialéctica indica que hay una mutua realimentación: desde lo manifiesto se llega a lo latente, lo latente es reenviado al nivel manifiesto para hacerse explícito, lo explicativo produce un insight que a su vez modifica la situación latente - por ejemplo, reduciendo los miedos básicos - y así sucesivamente va cumpliéndose la tarea a través de esta continua realimentación.

Las categorías de análisis de la interacción grupal, llamadas también vectores, son una escala de investigación y de evaluación tanto de procesos grupales como de los procesos individuales, utilizados para evaluar las dinámicas y los procesos de grupo. Estos procesos, son por los que pasa un grupo de trabajo centrado en la tarea, en este caso en la tarea de un desarrollo, no obstante estos son fenómenos universales representados con el siguiente esquema:

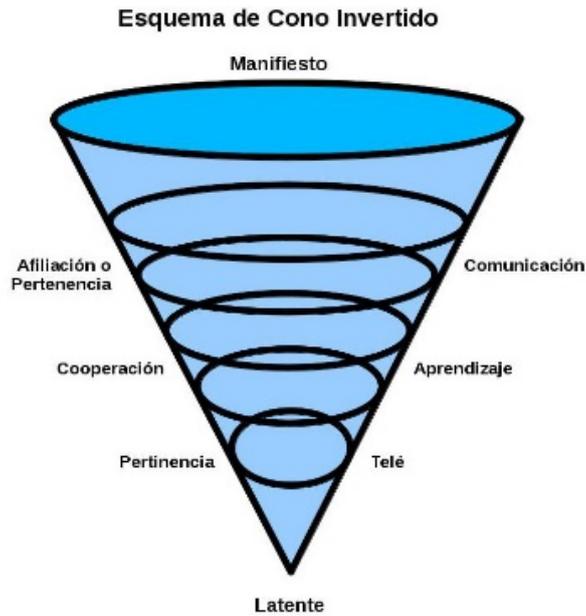


Figura 8. Esquema del cono invertido
Fuente: Pichón Rivière, El Proceso Grupal, página 230.

La Figura 8 permite comprender que la afiliación y la pertenencia, constituyen el grado de ligazón que los integrantes tienen entre sí y con la tarea.

La cooperación es la capacidad de articularse con otros para realizar una tarea; es la contribución, aún silenciosa, a la tarea grupal, sobre la base de la existencia de roles diferenciados

La pertinencia es la capacidad de centrarse en la tarea prescrita o sea es el involucramiento.

La comunicación es un contexto que incluye un mundo de señales que todos aquellos que se intercomunican saben codificar y decodificar de la misma manera.

La comunicación que se da entre los miembros, la que puede ser verbal o preverbal, a través de gestos.

Por su parte, el aprendizaje es la apropiación instrumental de la realidad para transformarla y transformarse y la telé es la disposición positiva o negativa para trabajar con un miembro del grupo o con un grupo.

Por último Afiliación o Pertenencia, grado de ligazón que los integrantes tienen entre sí y con la tarea, es decir hace referencia a la cohesión grupal, o sea al compromiso.

Resumen

En este capítulo se introdujo la problemática del desarrollo del software y su evolución a lo largo de su ciclo de vida, se desarrollaron conceptos vinculados a la forma en que se construye, utilizando procesos instanciados en proyectos, donde las personas son fundamentales para obtener los resultados esperados. Se tomaron conceptos de la Psicología social, más específicamente

- La construcción de un sistema es un proceso de construcción de modelos.
- Los procesos de desarrollo se clasifican en definidos o predecibles y empíricos o adaptativos.
- Dentro de los procesos definidos se trabajó sobre el Proceso Unificado de Desarrollo, por ser este el referente en la industria de este tipo de procesos.
- Las metodologías ágiles implementan la teoría de los procesos empíricos o adaptativos. Se seleccionó a SCRUM y XP dentro del conjunto de métodos ágiles, por ser éstos los más utilizados.
- SCRUM como método de gestión de proyectos y XP con el aporte de sus prácticas de ingeniería de producto se complementan y son frecuentemente utilizados en forma conjunta.
- A lo largo de este capítulo, se ha abordado los conceptos: modelos, procesos, metodologías, ciclos de vida, observando especialmente la relación entre ellos y las personas que los crean, desarrollan y utilizan.
- Al desarrollar este capítulo se destaca especialmente la incidencia del capital humano, y su participación fundamental como analistas simbólicos (Reich, R., 1993). dado que trabajan con productos tan complejos y abstractos como es el software.
- Se utiliza el concepto de ECRO (Esquema Conceptual Referencial y Operativo), entendiendo este como un conjunto articulado de conocimientos, que proporciona líneas de trabajo para la comprensión de los procesos sociales, particularmente de los grupales, indagando las formas de interacción para lo cual se utiliza el esquema del cono invertido en el que se registran modalidades de interacción observables, particularmente dentro de los grupos operativos, para nuestro caso particular los grupos de desarrollo de software.

CAPITULO IV

Industria del Software en Argentina

Composición del sector Informático en la República Argentina

Para caracterizar los centros denominados maduros, se toman los datos obtenidos a partir de una encuesta on line¹². La encuesta fue respondida por 153 empresas que en conjunto emplean a 13764 personas.

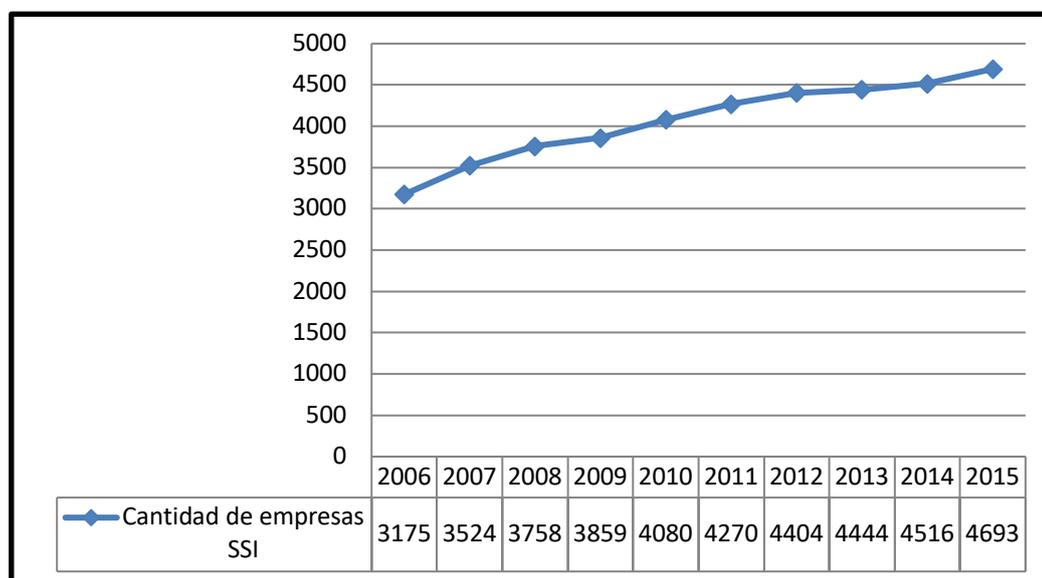


Gráfico 1. Evolución empresas SSI para el periodo 2006-2015

Fuente: Adaptado de Observatorio de Empleo y Dinámica Empresarial – MTTEy SS

A partir de los datos del Gráfico 1 se observa un significativo crecimiento del sector en los últimos diez años, ya que el número de empresas creció aproximadamente un 50 % de 3175 a 4693.

¹²Sistema de encuestas on line habilitado entre el 8 de febrero y el 09 de marzo de 2018. Observatorio Permanente de la Industria del Software y servicios Informáticos de la República Argentina (OPSSI) - Cámara de Empresas de Software y servicios Informáticos de la República Argentina (CESSI)

El mapa que se presenta ilustra la localización de los principales centros SSI en la República Argentina, diferenciándolos según su grado de desarrollo, en centros maduros, emergentes y nuevos actores, (Figura 9)

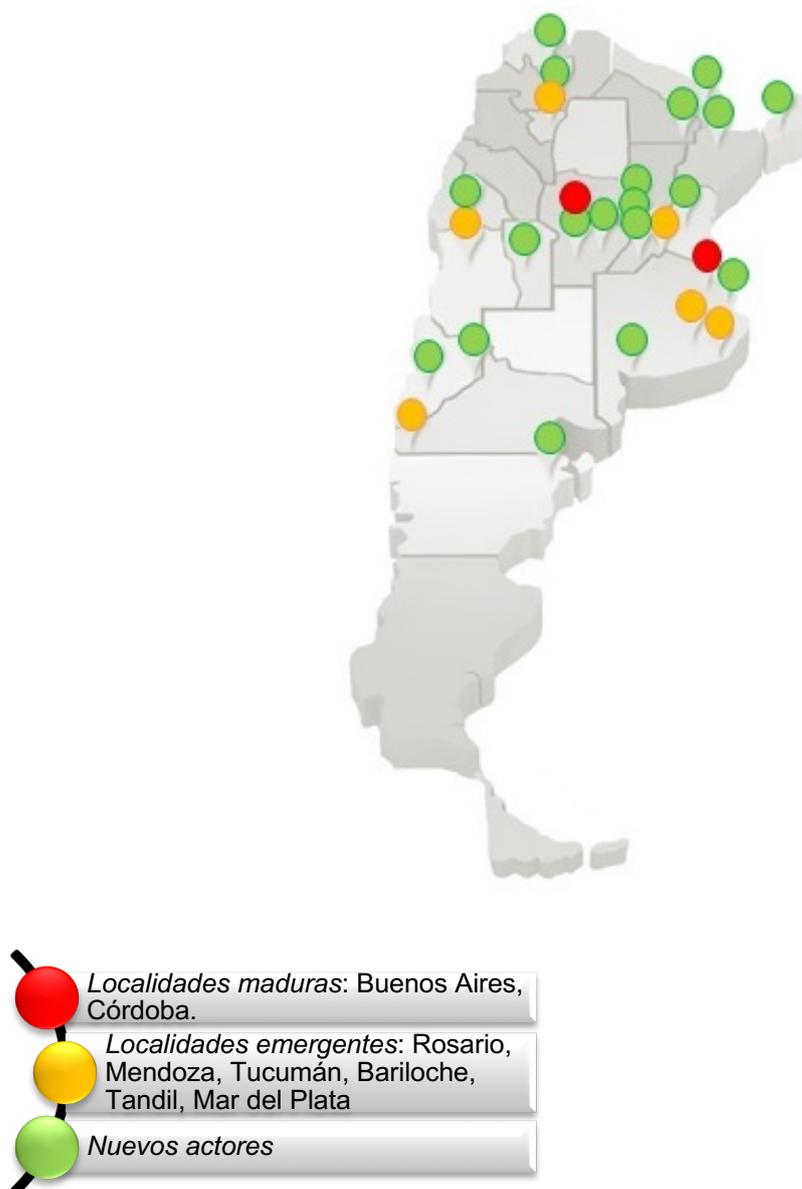


Figura 9. Localización Centros SSI de la República Argentina

Fuente: Programa de desarrollo Regional del Ministerio de Producción de la Nación. Empresas y Clústeres más competitivos

Cabe aclarar que se consideran empresas del sector SSI a las organizaciones privadas con trabajadores activos en relación de dependencia y que además la definición de su actividad principal ante organismos impositivos haya sido (Actividades de Informática exceptuando las de mantenimiento y reparación de equipos).

Si bien, por la metodología adoptada para la toma de datos, desde el punto de vista estadístico, no se puede afirmar que la muestra con la que trabaja el Observatorio sea estrictamente representativa del universo, - en la medida que las microempresas podrían estar subrepresentada,- se observa que su composición se aproxima a la del universo del sector en general.

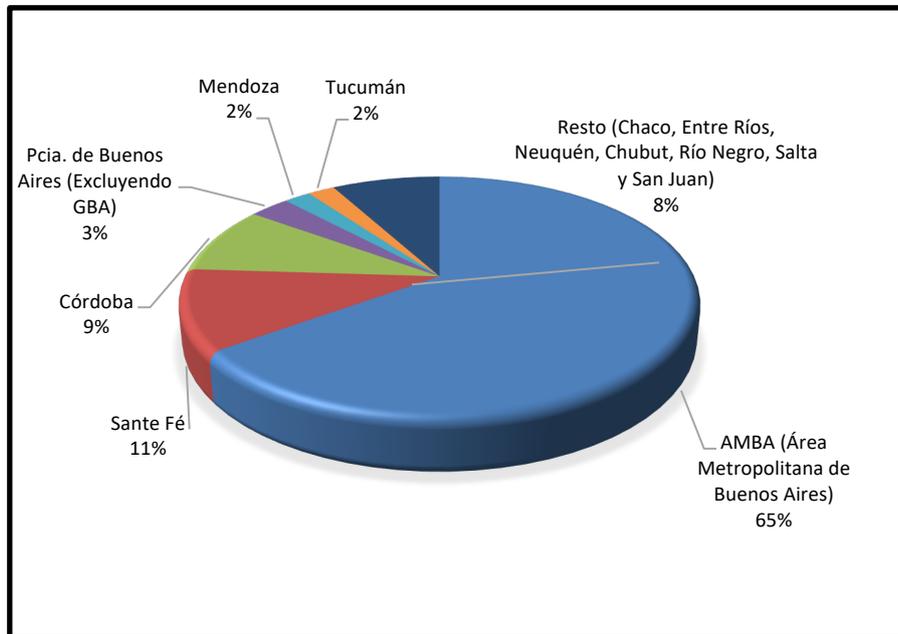


Gráfico 2. Empresas de Software en Argentina distribuidas por región

Fuente: Adaptado de datos del OPSSI-2018

De los datos obtenidos, (Gráfico 2) se observa que el 65% de las empresas se localiza en el AMBA (Área Metropolitana de Buenos Aires), un 11% en Santa Fe, un 9% en Córdoba, un 3% en provincia de Buenos Aires (excluyendo GBA), un 2% en Mendoza, un 2% en Tucumán, y el restante 8% se distribuye entre Chaco, Entre Ríos, Neuquén, Chubut, Río Negro, Salta y San Juan. Debido a estos resultados es que se caracteriza como localidades maduras a la ciudad de Córdoba y Ciudad Autónoma de Buenos Aires y localidades emergentes a Rosario, Mendoza y Tucumán.

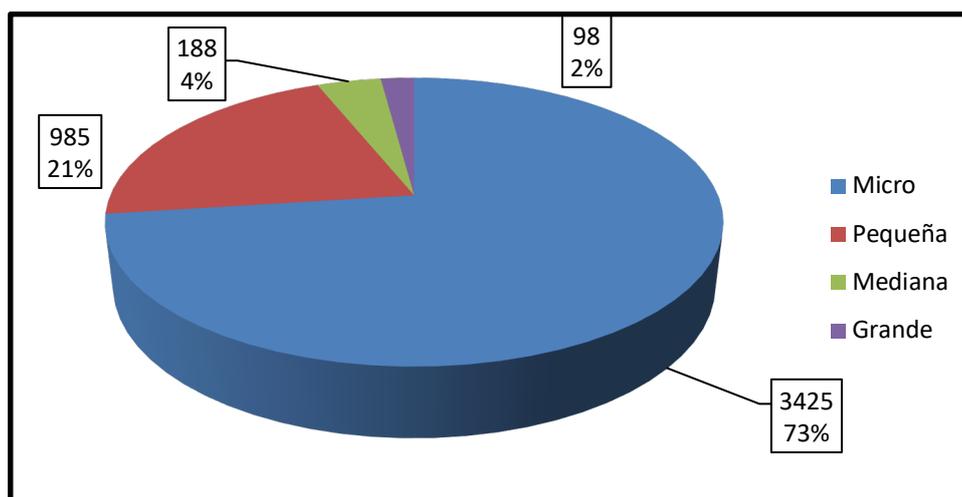


Gráfico 3. Tamaño de empresa en función de cantidad de empleados

Fuente: Adaptado de datos del OPSSI-2018

En el Gráfico 3 se puede observar el tamaño de las empresas en función de la cantidad de empleados.

A partir de los datos que provee OPSSI, en el año 2018 encontramos que el 73 % de las Empresas pueden caracterizarse como microempresas (hasta 9 trabajadores), y el 21% como pequeñas empresas (entre 10 y 49 trabajadores. Solo el 2%, pueden considerarse grandes Empresas (más de 200 trabajadores) y el 4 % Medianas Empresas (entre 50 a 200 trabajadores)

Ventas e Ingresos desde Exterior del Sector SSI

En el presente apartado se presentan como indicadores ventas totales e ingresos del exterior, a fin de caracterizar a grandes rasgos la evolución del sector en la última década.

En el Gráfico 4 y Gráfico 5 se presenta la evolución de ventas totales e ingresos desde el exterior en pesos corrientes y dólares estadounidenses respectivamente. En estos casos es importante tener en cuenta la devaluación que sufrió el peso con respecto al dólar estadounidense, tanto en enero de 2014 (50% anual) como en diciembre de 2015 (59 % anual durante 2016).

Se observa que por las devaluaciones, las ventas totales medidas en dólares estadounidenses cayeron alrededor del 11,1% en el año 2014 en tanto que en el año 2016 la caída fue del orden del 20,2%, en tanto que la estimación en pesos muestra un crecimiento significativos de 33,7 % y 27,2 % respectivamente.¹³

¹³ La estimación de ventas corresponde a relevamiento OPSSI en tanto que Ingresos desde el Exterior a la Balanza de Pagos INDEC.

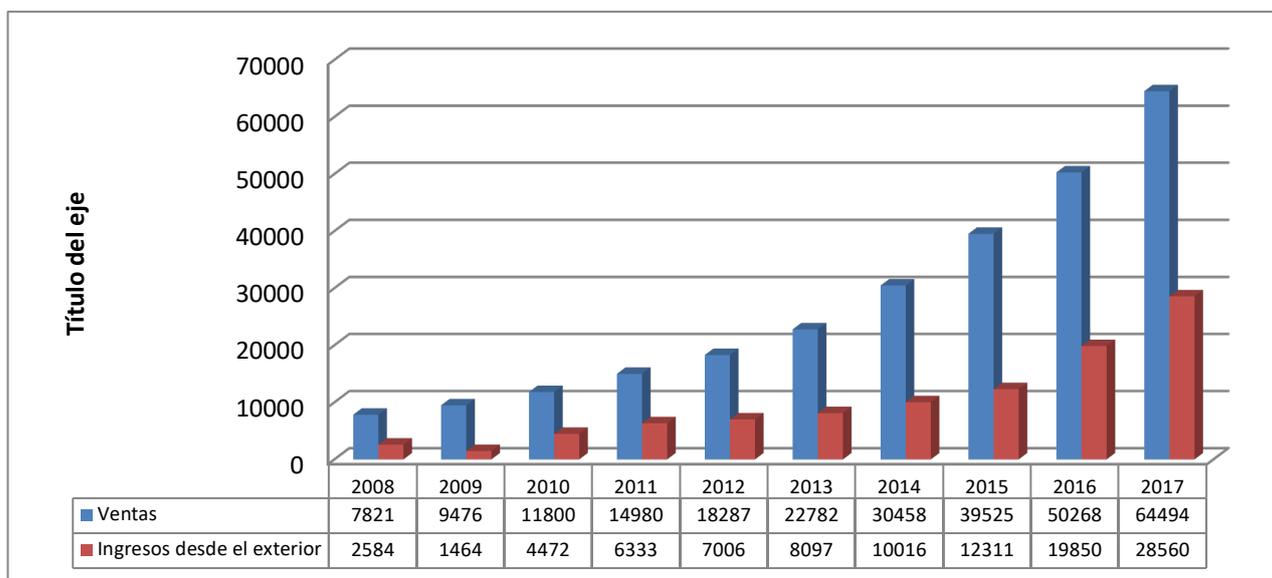


Gráfico 4. Evolución anual de ventas totales (en millones de pesos corrientes) e Ingresos desde el exterior (en millones de pesos corrientes) 2008-2017
Fuente: Adaptado de datos del OPSSI-2018

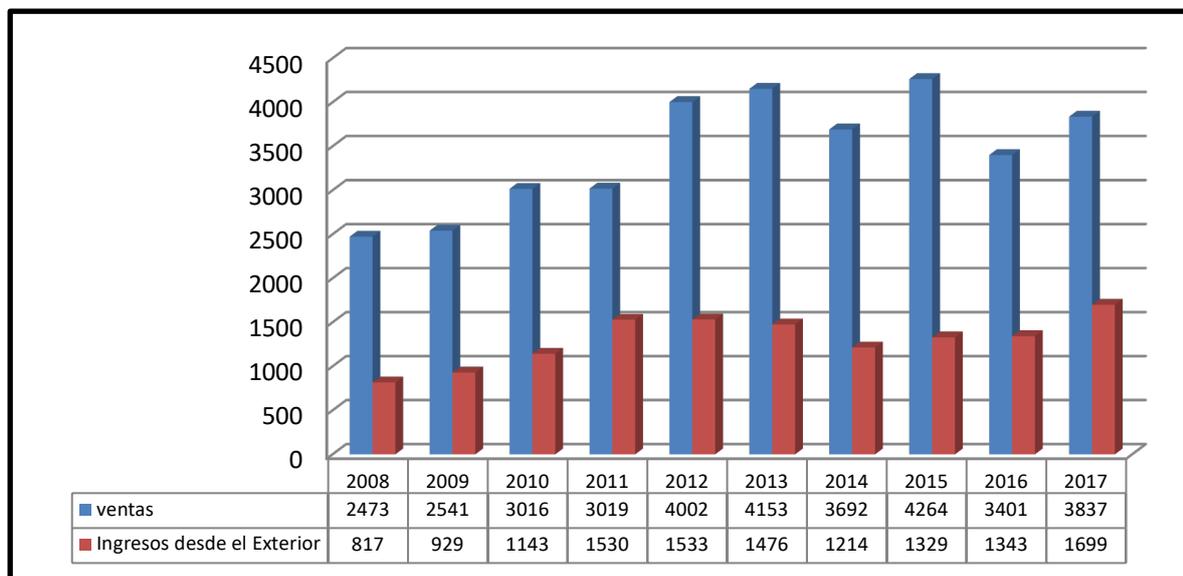


Gráfico 5. Evolución anual de ventas totales (en millones de USD) e ingresos desde el exterior (en millones de USD)
Fuente: Adaptado de datos del OPSSI-2018

Para dimensionar el peso del sector en la economía argentina, referimos estimaciones del OPSSI¹⁴, que estima que el empleo en las empresas del sector ha crecido un 42,2 % entre los años 2008 y 2017^a a una tasa acumulativa del 4%. En tanto que según datos del Observatorio de Empleo y Dinámica Empresarial del Ministerio de Trabajo, señala que para ese periodo el empleo registrado de todo el sector privado creció un 10,7% a una tasa anual acumulativa del 2,1% (OPSII, 2018). En este mismo sentido y a pesar de los efectos de la crisis mundial de los años 2008 y 2009 y las devaluaciones señaladas de 2014 y 2015 se observa que las ventas del sector medidas en moneda estadounidense aumentaron en los últimos diez años un 5% acumulativo anual. En tanto que los ingresos desde el exterior crecieron durante los últimos 10 años a una tasa anual del 8,5 % lo que resulta significativamente superior al compararlo con la cuenta de servicios de todo el país, que solo creció un 2,4% cumulado anual, según los registros de la balanza de pagos del INDEC.

Ventas por Actividad

Si se observa el Gráfico 6, se puede advertir que el desarrollo de software a medida y venta de productos propios e implementación e integración asociados a estos productos, están explicando el 65% de las ventas del sector durante el último año.

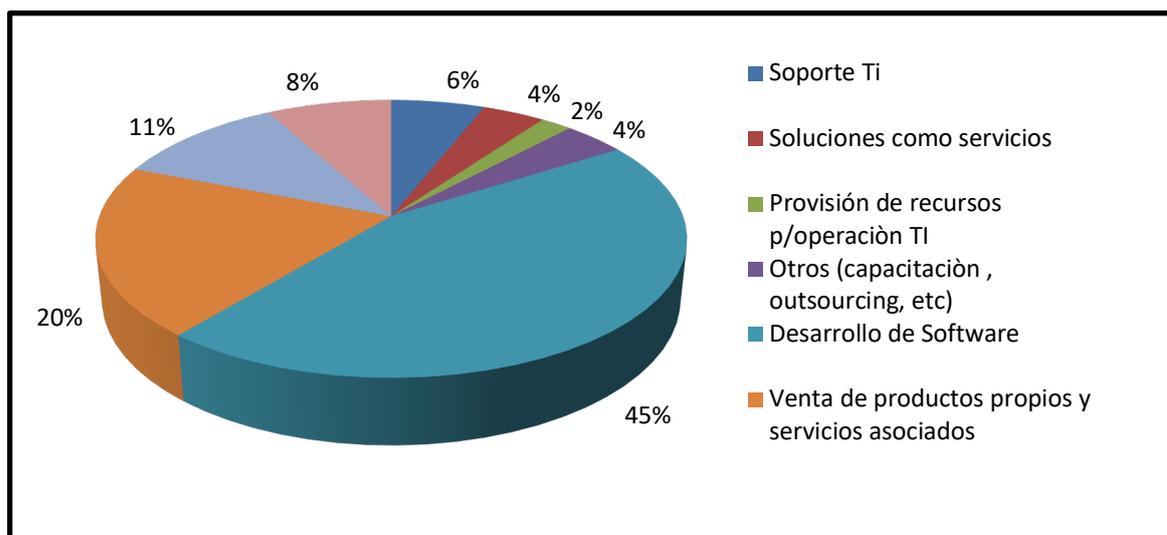


Gráfico 6. Participación de las distintas actividades en el total de ventas – Año 2017

Fuente : Adaptado de Datos del OPSSI-2018

¹⁴ Reporte anual sobre el Sector del Software y Servicios Informáticos de la República Argentina- Abril 2018

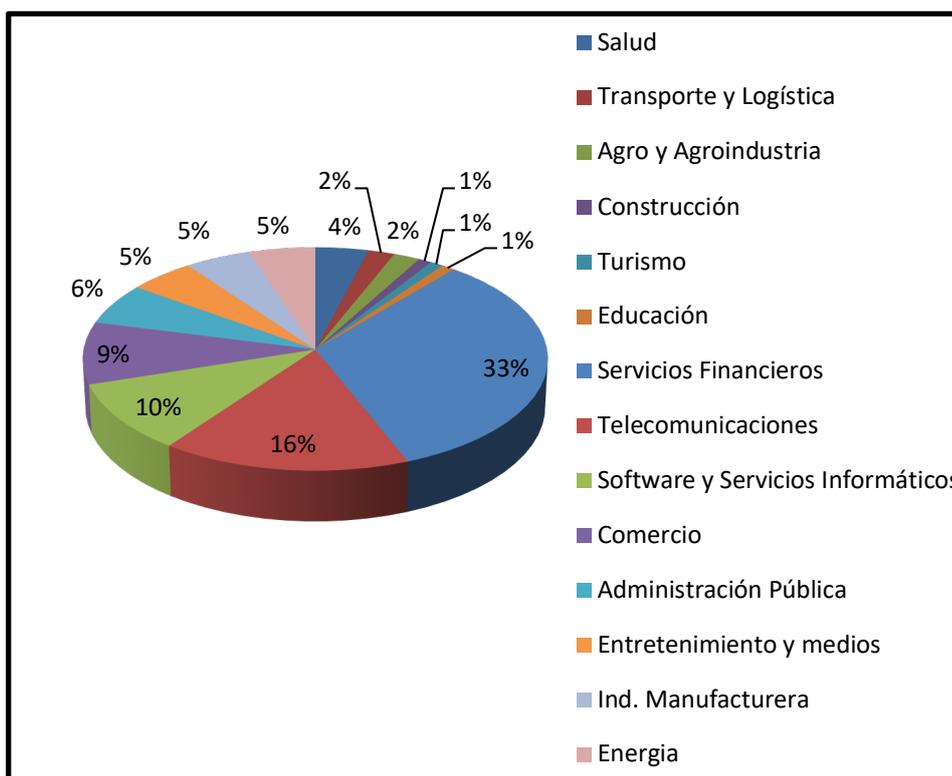


Gráfico 7. Principales clientes por sector y participación en las ventas. Año 2018
Fuente: Adaptado de datos del OPSSI-2018

Al analizar la distribución de las ventas del sector SSI entre sus clientes discriminados por sectores de actividad se observa que los servicios financieros, los que incluyen tanto bancos, como aseguradoras y servicios de pago electrónico, entre otras, son los principales clientes del sector, ya que representan un tercio del total facturado. A bastante distancia de esa posición, el sector telecomunicaciones se ubica en segundo lugar con un 16% del total de facturación, el sector SSI, quien se autoabastece a si mismo con un 10% de la facturación y en 4ª lugar se ubica el sector comercio con un 9%. (Ver Gráfico).

Ingresos del Exterior por Actividad

Si se analizan los ingresos desde el exterior por actividad, se observa que el Desarrollo de Software (56%) y la venta de productos propios y servicios asociados (25%), explican la mayor parte de los ingresos desde el exterior (81%). (Ver Gráfico 8)

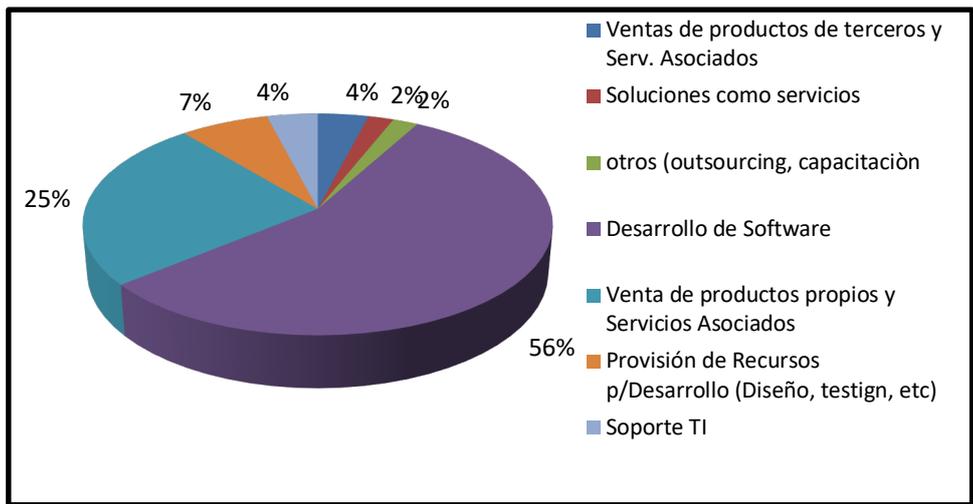


Gráfico 8. Participación de las actividades en los ingresos desde el exterior- 2018
 Fuente: Adaptado de datos del OPSSI-2018

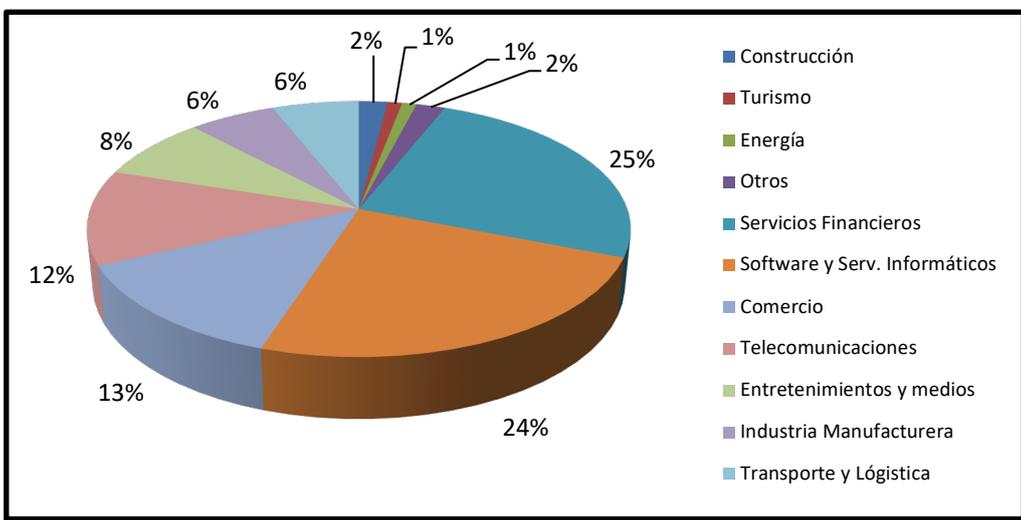


Gráfico 9. Principales clientes por sector y participación en los ingresos del sector – 2018
 Fuente: Adaptado de datos de OPSSI-2018
Si se observa el

Gráfico 9 se advierte que el sector financiero (25%) y las empresas de SSI (24%) representan casi el 50% de los ingresos desde el exterior.

Investigación, Desarrollo e Innovación

Por último para caracterizar la situación del sector en Argentina, se analizan las inversiones realizadas durante el año 2017 y los principales objetivos de I+D+i a los que las empresas destinaron su inversión.

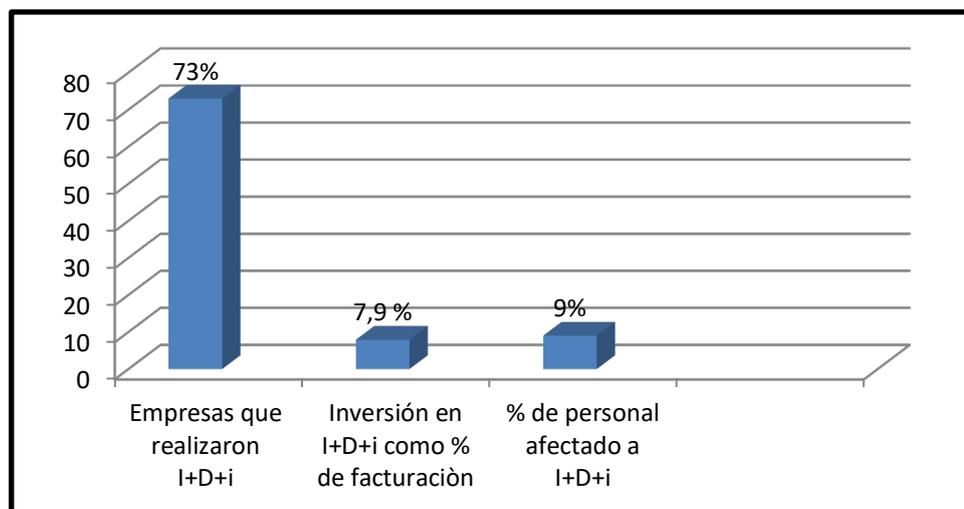


Gráfico 10. I+D+i de empresas del sector correspondiente al año 2017

Fuente: Adaptado de OPSSI-2018

Se observa que el 73% de las empresas durante el año 2017 ha realizado inversiones de I+D+i, con una inversión, que representa más del 7% de su facturación, afectando en promedio a 9% del total de su dotación de empleados. (Ver Gráfico 10)

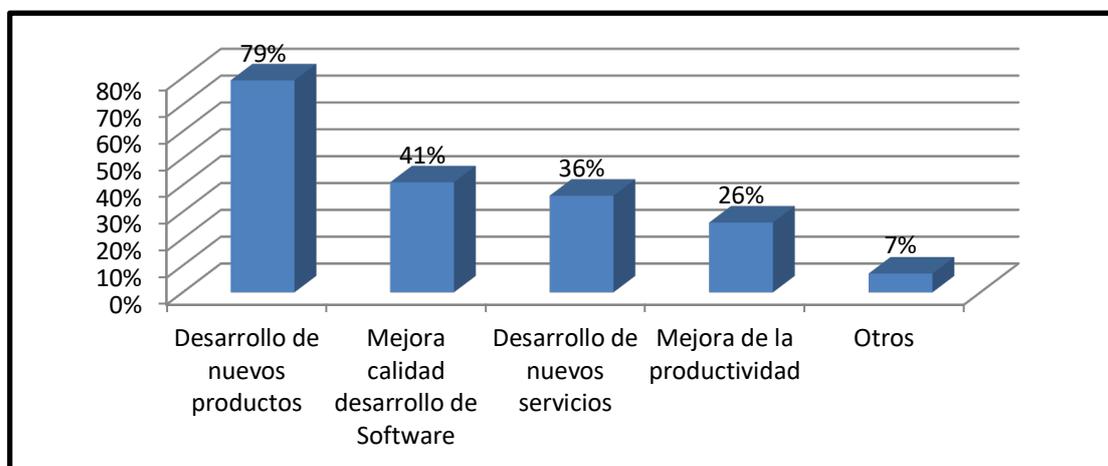


Gráfico 11. Principales objetivos de I+D+i – 2017

Fuente: Adaptado de OPSSI 2018

Si se analizan los datos del Gráfico 11 se advierte que por lejos la principal inversión del sector en I+D+i ha sido en el desarrollo de nuevos productos (79%), en tanto que la mejora de la calidad de desarrollo de Software con 41% y el desarrollo de nuevos servicios con 35% , le siguen en la lista de objetivos de las empresas. La mejora de la productividad también ha resultado importante ya que se registra que un 25% de la inversión ha sido derivada a dicho objetivo.

Estos datos, estarían dando sustento a la justificación que planteamos para el presente trabajo de tesis, en la medida del interés y preocupación de las industrias del sector para expandir sus productos y servicios al tiempo que mejorar la calidad y productividad de sus empresas.

CAPÍTULO V

Perspectivas Metodológicas del Estudio

Alcances del Diseño

El presente estudio, adopta un diseño fundamentalmente cuantitativo con un abordaje cualitativo para indagar un aspecto puntual en relación con la perspectiva de género en el sector informático.

La selección de estrategias metodológicas ha tenido como fin último dotar a la investigación de validez y confiabilidad. Estas estrategias han sido estructuradas en fases con el fin de exponer con claridad los objetivos que cada una de ellas persiguió y además facilitar la explicación de los procedimientos e instrumentos utilizados. Cabe aclarar que esta división no implica una secuenciación temporal lineal, ya que muchas de las actividades se superponen en el tiempo que demandó la realización del presente estudio.

Las estrategias, que en relación con el abordaje cuantitativo desarrollaron se pueden identificar como:

- a) Análisis Descriptivo
- b) Análisis Bivariado
- c) Análisis Multivariado
 - 1. Matriz de correlación Pearson
 - 2. Análisis Factorial de Correspondencias Múltiples

La estrategia cualitativa que se desarrolló fue:

- a) Focus Group

Análisis Descriptivo

Los estudios descriptivos buscan especificar las propiedades importantes de las personas, grupos, comunidades o cualquier otro fenómeno que sea sometido a análisis (Hernández Sampieri: 2014). En líneas generales podemos decir que los estudios descriptivos miden de manera más bien independiente los conceptos o variables a los que se refieren, aunque el autor señala también que las mediciones de cada una de dichas variables se pueden integrar para decir cómo es y cómo se manifiesta el fenómeno de interés. (Hernández Sampieri: 2014).

Se trata de estudios observacionales, en los cuales no se interviene o manipula el factor de estudio, es decir se observa lo que ocurre con el fenómeno en estudio en condiciones naturales, en la realidad.

A su vez sabemos que también pueden ser clasificados en transeccionales y longitudinales. Los estudios transeccionales intentan analizar el fenómeno en un punto en el tiempo, por eso también se los denomina “de corte”; los estudios longitudinales en cambio se dedican a estudiar uno o más factores en un periodo de tiempo más largo.

En el presente estudio esta fase tuvo fundamentalmente un objetivo diagnóstico al intentar caracterizar las principales características de los equipos de trabajo que se desempeñan en el sector Informático, a través de la toma de datos en un momento determinado. Por esta razón se trata de un estudio transeccional. La obtención de datos se realizó en principio desde el propio marco teórico y como se señaló, desde una aproximación cuantitativa, basada en la investigación por encuestas dirigidas profesionales del sector.

Análisis Bivariado

Una variable cualitativa bidimensional está dada por dos atributos que se observan simultáneamente sobre los individuos de una población. De forma análoga al caso de dos variables numéricas, la distribución de frecuencias conjunta una variable cualitativa bidimensional (A, B) está definida por los pares de datos observados sobre los individuos de la población junto con sus frecuencias absolutas. Los datos pueden organizarse en serie o en una tabla de doble entrada. La tabla de doble entrada para caracteres cualitativos recibe el nombre de tabla de contingencia o tabulación cruzada.

Si organizamos los datos en una tabla de doble entrada, entonces mostraremos, por ejemplo, las modalidades del atributo A (valores distintos de A) por filas en la primera columna de la tabla

(A1, A2, . . . ,Ak), las modalidades del atributo B (valores distintos de B) por columnas en la primera fila de la tabla (B1, B2, . . . , Bp), y las cantidades n_{ij} en el interior de la tabla indican el número de individuos de la población que presentan simultáneamente la modalidad i -ésima de A y la modalidad j -ésima de B, esto es, la frecuencia absoluta del par de valores (Ai, Bj).

Así la representación típica de una tabla de contingencia $k \times p$ (k filas y p columnas) es:

A \ B	B ₁	B ₂	...	B _j	...	B _p	
A ₁	n_{11}	n_{12}	...	n_{1j}	...	n_{1p}	
A ₂	n_{21}	n_{22}	...	n_{2j}	...	n_{2p}	
⋮	⋮	⋮		⋮		⋮	
A _i	n_{i1}	n_{i2}	...	n_{ij}	...	n_{ip}	
⋮	⋮	⋮		⋮		⋮	
A _k	n_{k1}	n_{k2}	...	n_{kj}	...	n_{kp}	
							n

En este caso, el número total de individuos de la población, n, es la suma de todas las frecuencias n_{ij} del interior de la tabla.

Independencia:

Diremos que los atributos A y B son independientes si la proporción de individuos que presentan conjuntamente los valores (A_i, B_j) de (A, B) entre los que presentan el valor A_i de A es la misma para cualquier valor de j ; o equivalentemente, la proporción de individuos que presentan conjuntamente los valores (A_i, B_j) de (A, B) entre los que presentan el valor B_j de B es la misma para cualquier valor de i .

Entonces, dos atributos A y B son estadísticamente independientes si y solo si

$$n_{ij} = \frac{n_i \cdot n_j}{n}, \quad \forall i = 1, 2, \dots, k; j = 1, 2, \dots, p$$

Prueba de Chi cuadrado de independencia

Existe un contraste formal para la hipótesis nula de independencia de los atributos A y B aun determinado nivel de significación α , a partir de la información muestral recogida en la tabla de contingencia. La hipótesis alternativa es la existencia de asociación entre los atributos A y B.

H0: A y B son independientes

H1: A y B no son independientes

Este test es conocido como test Chi-cuadrado y se basa en la distribución bajo la hipótesis nula del llamado coeficiente de contingencia χ^2 (coeficiente Chi-cuadrado), que se expresa como:

$$\chi^2 = \sum_{i=1}^k \sum_{j=1}^p \frac{(n'_{ij} - n_{ij})^2}{n'_{ij}}$$

Podemos determinar la probabilidad que deja a la derecha el valor del estadístico χ^2_{exp} en una distribución $\chi^2_{(k-1)(p-1)}$, conocida como p-valor de contraste. Así $\chi^2_{exp} > \chi^2_{(k-1)(p-1), \alpha}$ si y solo si p-valor $< \alpha$.

Por lo tanto,

Valor $p \leq \alpha$: Las variables tienen una asociación estadísticamente significativa (Rechazar H_0)

Si el valor p es menor que o igual al nivel de significancia, usted rechaza la hipótesis nula y concluye que hay una asociación estadísticamente significativa entre las variables.

Valor $p > \alpha$: No se puede concluir que las variables están asociadas (No se puede rechazar H_0)

Si el valor p es mayor que el nivel de significancia, usted no puede rechazar la hipótesis nula, porque no hay suficiente evidencia para concluir que las variables están asociadas.

Análisis Multivariado

Coeficiente de correlación Pearson (r)

El coeficiente de correlación de Pearson (r) es una prueba estadística para analizar la relación entre dos variables.

La matriz de correlación muestra los valores de correlación de Pearson, que miden el grado de relación lineal entre cada par de elementos o variables. Los valores de correlación se pueden ubicar entre -1 y +1, donde el signo negativo o positivo es el que muestra el tipo de correlación negativa o positiva respectivamente.

La hipótesis de investigación señala el nivel de significancia de la correlación sin evaluar la causalidad. Si bien, la noción causa-efecto (independiente-dependiente) es posible establecerla teóricamente, es importante aclarar que la prueba no asume dicha causalidad. Las relaciones que se pueden establecer a partir de la prueba son del tipo “a mayor X, mayor Y”, “a mayor X, menor Y”, “altos valores en X están asociados con altos valores en Y”, “altos valores en X se asocian con bajos valores de Y”. (Sampieri, et al , 2014)

Análisis Factorial de Correspondencias Múltiples

Es una técnica de análisis factorial que

- Busca factores que son la expresión de combinaciones de las variables originales
- Expresan los principales factores de diferenciación (inercia explicada)
- Ordenados jerárquicamente
- Reducen los datos: “Pérdida de información y ganancia en significación”
- La representación gráfica es una ayuda para la interpretación: la proximidad en el espacio significa “correlación” entre categorías: correspondencias

En el Gráfico 12 se observa la relación entre las variables originales y los factores, así como su representación gráfica

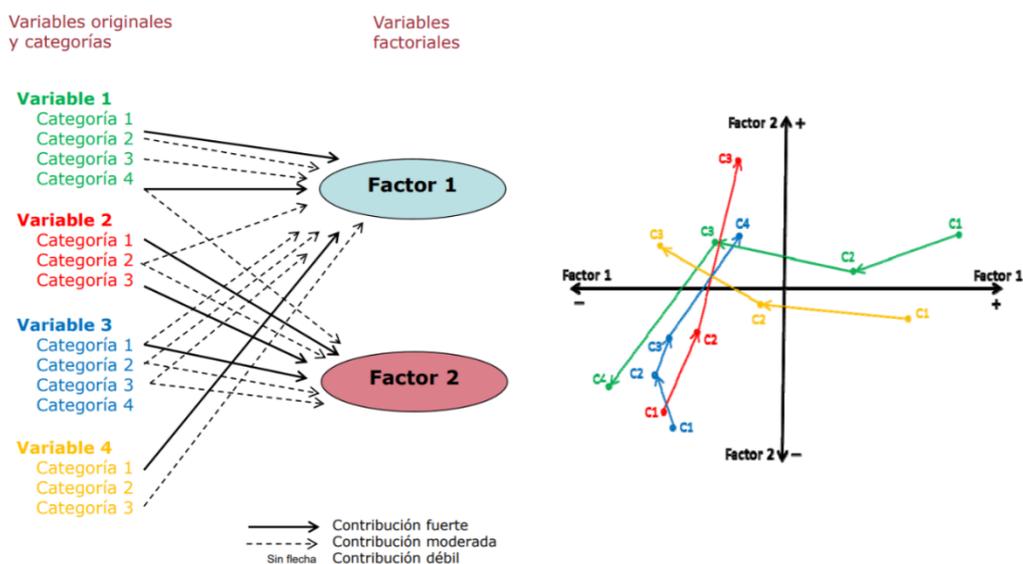


Gráfico 12. Relación entre las variables originales y los factores, así como su representación gráfica. Fuente: López- Roldan , P. &Fachillas. Análisis de correspondencias, Seminario de doctorado UBA¹⁵.

El Análisis de Correspondencias trata de analizar, describir y representar gráficamente la información contenida en una tabla de distribución conjunta de datos dispuestos en filas y columnas: sus correspondencias (asociaciones). En la representación gráfica cada categoría o valor de la variable se representa como un punto en el espacio: puntos-fila y puntos-columna. Las proximidades geométricas entre puntos-fila y puntos-columna traducen las asociaciones estadísticas entre filas y columnas.

Instrumento de Recolección de Datos para la Estrategia Cuantitativa

Definiciones operacionales de la Encuesta

A continuación se presentan la contextualización formulada en la Encuesta que se encuentra agregada como ANEXO I y que fue administrada a 57 profesionales del sector cuyo desempeño se da fundamentalmente vinculado a los denominados centros maduros.

1. Edad

Esta pregunta se incluye, por ser este un factor diferencial en esta profesión dado que el envejecimiento técnico es más impiadoso que en otras debido a la creciente evolución de las herramientas tecnológicas que deben utilizarse y que acompañan al profesional a lo largo de toda su carrera en mayor o menor proporción. El profesional informático nunca deja de abordar un importante porcentaje de conocimiento técnico a lo largo de su carrera.

2. Sexo

¹⁵López- Roldan , P. &Fachelli,S.(2015). Análisis de correspondencias, Seminario de doctorado UBA. En: <http://pagines.uab.cat/plopez/sites/pagines.uab.cat.plopez/files/ACO-UBA.pdf>. [Consultado 01/10/2018]

Esta pregunta se incluye, en función a la hipótesis de la selección de ciertos roles por cada sexo y a una legítima preocupación actual. De la simple observación de la conformación de las dotaciones de recursos humanos en las empresa, se observa una escasa presencia de mujeres en la profesión; circunstancia esta que es también corroborada por la propia comunidad informática. Situación que no siempre fue así.

3. Formación o Título

En este caso las opciones fueron:

1. Ingeniero en Sistemas de Información o similar
2. Licenciado en Informática o similar
3. Analista Universitario de Sistemas
4. Otro, indique cuál

La profesión informática se ha incorporado a las profesiones incluidos dentro del artículo 43, bajo la Res. ME 786 del año 2009, identificando 5 terminales, 2 ingenierías y 3 licenciaturas: Ingeniería en Computación, Ingeniería en Sistemas de Información/Informática, Licenciatura en Sistemas de Información, Licenciatura en Informática y Licenciatura en Ciencias de la Información. Cabe agregar que en la mayoría de las universidades donde se imparten estas carreras, también se expide el título intermedio de Analista Universitario de Sistemas.

La opción "Otro", está incluida esencialmente debido a la existencia de un importante número de idóneos, que teniendo otros títulos u otras profesiones han desarrollado su carrera como informáticos.

4. Localidad donde trabaja

Si bien una de las opciones de respuesta fue abierta (otra, cual) , teniendo en cuenta el perfil de localidades /empresas con el que se trabajó, se desagregaron las respuestas para Ciudad Autónoma de Buenos Aires y Córdoba, ya que delimitar con precisión ese perfil fue uno de los objetivos del trabajo.

1. Ciudad Autónoma de Buenos Aires (CABA)
2. Provincia de Buenos Aires
3. Córdoba
4. Rosario
5. Otra, cuál

En particular, se han seleccionado las localidades de Ciudad de Buenos Aires y Ciudad de Córdoba, por ser las únicas localidades consideradas por los estudios de Servicios Basados en Conocimiento, como localidades maduras, esto radica que en esta profesión es de fundamental importancia la formación del profesional y en estas dos localidades hay un importante número de casas de altos estudios que dan formación a esta disciplina. Adicionalmente son en estas localidades, donde se advierte una mayor concentración de la industria vinculada al sector que se analiza.

5. Actividad Laboral

La pregunta relacionada con esta dimensión es abierta, dado que se quiso conocer el espectro actual del mercado, que por cierto siempre resulta cambiante. La expectativa fue que el encuestado describa su principal función/rol/posición de trabajo dentro del mercado laboral.

6. Antigüedad en la última función de trabajo

En esta pregunta se consultó por la última función de trabajo debido a la alta volatilidad que la actividad presenta en los puestos de trabajo. Esta circunstancia está dada por la alta demanda laboral, un importante crecimiento vegetativo y acciones como el *hunting* (búsqueda y selección para cubrir cargos de altos mandos), *recruiting* (reclutamiento y búsqueda de puestos laborales noveles, iniciándose en mundo laboral de los servicios basados en el conocimiento) y *searching* (búsqueda de puestos técnicos de alto nivel). Estos términos son utilizados frecuentemente en las áreas de búsqueda y selección de capital intelectual, sobre todo para áreas técnicas, tal como es considerado el sector económico de la industria del software.

7. ¿El equipo de desarrollo en el que trabaja, está co-locado o distribuido?

Las posibles respuestas fueron:

1. Co-locado
2. Distribuido
3. Otra modalidad, ¿cuál?

Una de las características del trabajo en equipo en las áreas de sistemas de información/informática, es que los equipos de trabajo pueden estar distribuidos, es decir el equipo puede trabajar geográficamente disperso. El concepto de equipo de trabajo se da porque el grupo es mediado por las tecnologías de información y comunicación. Esta distribución también puede darse en países con diferentes husos horarios.

Por otro lado, cuando se alude a equipos co-locados, se hace referencia a equipos que trabajan físicamente en el mismo ambiente.

Cabe aclarar que también puede darse un esquema mixto de vinculación.

8. ¿Cuáles son las tres tareas principales que usted desarrolla en su trabajo?

Las alternativas de respuestas fueron:

1. Gestión de Proyecto
2. Ingeniería de Requerimientos
3. Arquitectura
4. Diseño

5. Implementación
6. Testing
7. Despliegue
8. ¿Otras? ¿Cuáles?

Estas actividades son, en términos generales, las involucradas en el proceso de desarrollo de software, que si bien han sido descritas en forma genérica en el marco teórico, en esta oportunidad se describen ad hoc a los fines de facilitar la interpretación de las respuestas y de dar sentido a la pregunta.

Gestión de Proyecto: Esta disciplina es la responsable de administrar el proyecto, incluyendo la planificación y el seguimiento de la ejecución de este. Se entiende por proyecto la unidad de administración de recursos y personas involucradas en la creación de un sistema informático. Quienes trabajan realizando esta actividad, suelen cubrir roles de Líder de Proyecto/Jefe de Proyecto.

Ingeniería de Requerimientos: Esta disciplina es la encargada de la definición de todas las capacidades y características que el producto de software a construir debe reunir para satisfacer las expectativas y necesidades de los clientes/usuarios. Implica las actividades de descubrimiento, especificación, validación de los requerimientos y la gestión de cambios que puedan presentarse sobre estos.

Arquitectura: Es una disciplina de diseño de alto nivel, responsable de la aplicación del modelo de requerimientos a una tecnología específica que solucione el caso específico.

Diseño: Esta disciplina técnica toma el diseño estratégico de alto nivel que produce la arquitectura y decide sobre los detalles tácticos.

Implementación: La actividad de implementación hace referencia a la aplicación del modelo de diseño con un lenguaje de programación. Esta función suele llamarse también programación, desarrollo o construcción.

Prueba (testing): Esta actividad tiene el propósito de realizar la validación y verificación del producto de software construido para determinar su adecuación a los requerimientos y su correcto funcionamiento.

Despliegue: Esta disciplina es la encargada de la instalación del producto de software en el ambiente de producción del cliente. También se lo conoce como puesta en producción o implantación.

Otras: Se trató de una pregunta abierta para permitir a los encuestados describir las actividades que no están incluidas en alguna de las opciones anteriormente descritas.

9. En el ámbito del equipo de desarrollo en el que trabaja, utiliza metodologías basadas en procesos definidos. Si es así ¿cuáles?

Las opciones de respuesta para este ítem fueron:

1. PUD (Proceso Unificado de Desarrollo)
2. RUP (Rational Unified Process)
3. ICONIX
4. Yourdon
5. Microsoft Solution Framework
6. No utilizo metodologías basadas en procesos definidos
7. ¿Otras? ¿Cuáles?

A continuación se describen en forma sintética cada una de las metodologías

PUD (Proceso Unificado de Desarrollo): Es la descripción de una metodología basada en procesos definidos. El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y además, por ser iterativo e incremental. Es la descripción de un entorno genérico de proceso cuyas actividades se adaptan a las necesidades de cada proyecto en particular.

RUP (Rational Unified Process): El Proceso Unificado de Rational o RUP (por sus siglas en inglés de Rational Unified Process) es un proceso de desarrollo de software creado por la empresa Rational Software, actualmente propiedad de IBM. Es una extensión y evolución del PUD. Se trata de una de las metodologías estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

ICONIX: Es un proceso de desarrollo de software simplificado en comparación con otros más tradicionales, que unifica un conjunto de métodos con el objetivo de abarcar todo el ciclo de vida de un proyecto. Se trata de una metodología pesada-ligera de Desarrollo del Software que se halla entre RUP (Rational Unified Process) y XP (eXtreme Programming), unifica un conjunto de métodos de orientación a objetos con el fin de tener un control estricto sobre todo el ciclo de vida del producto a realizar.

Yourdon: metodología basada en el paradigma estructurado denominada *Análisis Estructurado Moderno*. Determina un conjunto de tareas que deben realizarse para definir, especificar y construir un producto de software.

Microsoft Solution Framework: Es un conjunto de principios, modelos, disciplinas, conceptos y directrices para la prestación de servicios de tecnología de la información, propiedad de la empresa Microsoft.

No utilizo metodologías basadas en procesos definidos: esta opción de respuesta indica que el encuestado no trabaja con metodologías de desarrollo basadas en procesos definidos.

¿Otras? ¿Cuáles?: pregunta abierta para permitir a los encuestados describir la metodología utilizada en caso de no estar incluida en alguna de las opciones anteriormente descriptas.

10. En el ámbito del equipo de desarrollo en el que trabaja, utiliza metodologías/frameworks basados en filosofía Lean/Ágil, si es así ¿cuáles?

Las opciones de respuesta fueron las siguientes:

1. Scrum
2. XP
3. FDD
4. Kanban
5. Scrumban
6. ATDD/TDD
7. No utilizo metodologías basadas en metodologías Lean/Ágiles
8. ¿Otras?

A continuación se presentan las especificaciones técnicas de las alternativas dadas en la encuesta:

Scrum: Marco de trabajo que define prácticas para la Gestión Ágil de Proyecto, el mismo está basado en procesos empíricos de control.

XP: Es un marco de trabajo para el desarrollo de software basada en los valores de simpleza, comunicación, feedback y coraje. Reúne al equipo completo junto a prácticas simples, con el feedback suficiente para permitirle al equipo ver en dónde está y ajustar las prácticas a su situación única.

FDD: (Feature Driven Development): Se trata de un Desarrollo Basado en Funcionalidades, es un enfoque ágil para el desarrollo de sistemas. Como las otras metodologías adaptables, se enfoca en iteraciones cortas que entregan funcionalidad tangible. Este enfoque no hace énfasis en la obtención de los requerimientos sino en cómo se realizan las fases de diseño y construcción. FDD afirma ser conveniente para el desarrollo de sistemas críticos.

Kanban: El método Kanban es una aproximación al proceso gradual, evolutivo y al cambio de sistemas para las organizaciones. Utiliza un sistema de extracción limitada del trabajo en curso, como mecanismo básico para exponer los problemas de funcionamiento del sistema (o proceso) y estimular la colaboración para la mejora continua del sistema. Es un método basado en la filosofía lean.

Scrumban: La metodología Scrumban nace de la combinación de principios de los métodos ágiles de gestión de proyectos Scrum y Kanban. Aunque en principio pueden parecer muy similares, las dos estrategias de gestión presentan diferencias en la manera de ejecutar el proyecto. Es por eso por lo que el novedoso Scrumban se encarga de combinar aquellos elementos que resultan complementarios.

ATDD/TDD: (Acceptance Test-Driven Development), es una práctica en la que todo el equipo, destacando, e incluyendo, a los usuarios y/o al product owner, desarrolladores y tester, analiza conjuntamente los criterios de aceptación, antes de que comience el desarrollo. Es decir, que a diferencia del Test de Aceptación tradicional, el ATDD se diferencia en que las pruebas de aceptación se hacen antes de desarrollar y que además se automatizan.

TDD (Test-driven development): Desarrollo guiado por pruebas de software, es una metodología de desarrollo que implica escribir las pruebas primero (Test First Development) y Refactorización (Refactoring). El propósito del desarrollo guiado por pruebas es lograr un código limpio que funcione. La idea es que los requisitos sean traducidos a pruebas, de este modo, cuando las pruebas pasen se garantizará que el software cumple con los requisitos que se han establecido.

No utilizo metodologías basadas en metodologías Lean/Ágiles: Esta opción de respuesta indica que el encuestado no trabaja con metodologías de desarrollo basadas en filosofías lean/ágil.

¿Otras?: pregunta abierta para permitir a los encuestados describir la metodología utilizada en caso de no estar incluida en alguna de las opciones anteriormente descritas.

11. ¿Qué metodología de gestión de proyectos utiliza el equipo de desarrollo en el que trabaja?

Las opciones de respuesta fueron:

1. Metodología de gestión basada en PMI
2. Metodología de gestión basada en CMMI
3. Metodología de gestión ágil
4. Metodología de gestión lean
5. Metodología de gestión basada en RUP
6. No utilizo metodologías de gestión de Proyectos
7. Otras, ¿cuáles?

Metodología de gestión basada en PMI: Esta pregunta se orienta a identificar si los encuestados utilizan para gestionar sus proyectos prácticas derivadas de la Guía del PMBOK (del inglés Project Management Body of Knowledge), desarrollada por el Project Management Institute (PMI). Esta guía contiene una descripción general de los fundamentos de la Gestión de Proyectos reconocidos como buenas prácticas para lograr un gerenciamiento eficaz y eficiente del proyecto.

Metodología de gestión basada en CMMI: Esta pregunta estuvo orientada a identificar si los encuestados utilizan para gestionar sus proyectos prácticas derivadas del Modelo de Madurez y Capacidad Integrado (CMMI) perteneciente al CMMI Institute. Es un modelo de procesos que contiene las mejores prácticas de la industria para el desarrollo, mantenimiento, adquisición y operación de productos y servicios.

Metodología de gestión Ágil: Esta pregunta tuvo como finalidad identificar si los encuestados utilizan para gestionar sus proyectos, alguna metodología ágil, basada en las recomendaciones del Manifiesto Ágil.

Metodología de gestión Lean: Esta pregunta estuvo dirigida a identificar si los encuestados utilizan para gestionar sus proyectos, alguna metodología lean, basada en las recomendaciones y principios de lean.

Metodología de gestión basada en RUP: Esta pregunta tuvo como objetivo identificar si los encuestados utilizan para gestionar sus proyectos, alguna metodología basada en el Proceso Unificado de Rational, proceso definido con anterioridad.

No utilizo metodologías de gestión de Proyectos: Esta pregunta tuvo como fin identificar si los encuestados utilizan para gestionar sus proyectos alguna metodología de las conocidas o si sigue simplemente un método secuencial de construcción.

¿Otras? ¿Cuáles?: Se incluyó una pregunta abierta para permitir a los encuestados describir la metodología utilizada en caso de no estar incluida en alguna de las opciones anteriormente descritas.

12. ¿En los equipos de desarrollo de software en los cuáles trabajó o trabaja actualmente, qué competencias técnicas son requeridas? Puede marcar más de una.

1. Programación o Desarrollo
2. Manejo de Técnicas de Ingeniería de Requerimientos (Casos de Uso, Técnicas de Elicitación)
3. Gestión ágil de requerimientos (user stories)
4. Testing de Software
5. Gestión de Personas
6. Gestión Ágil de Proyectos
7. Gestión Tradicional de Proyectos
8. Modelado de Arquitecturas de Software
9. Diseño de Interacción Humano-Máquina
10. Manejo de Herramientas para Testing Automatizado
11. Manejo de Herramientas para Gestión de Configuración/Integración Continua
12. Otras, ¿cuáles?

A continuación se incluye la descripción conceptual de las opciones.

Programación o Desarrollo: Esta opción de respuesta se refiere a competencias para programación de software sin hacer distinción del lenguaje de programación utilizado.

Manejo de Técnicas de Ingeniería de Requerimientos (Casos de Uso, Técnicas de Elicitación): Esta opción de respuesta hace referencia a competencias técnicas para el desarrollo de requerimientos en el contexto de la ingeniería de requerimientos. Esto implica el descubrimiento, especificación y validación de los requerimientos para el producto de software.

Gestión ágil de requerimientos (user stories): Esta opción de respuesta hace referencia a competencias técnicas para la gestión de requerimientos en ambientes ágiles, utilizando técnicas adecuadas como la de user stories.

Testing de Software: La opción de respuesta hace referencia a competencias técnicas para la realización de pruebas al software.

Gestión de Personas: La opción de respuesta refiere a actividades relacionadas con la administración de las personas que integran los equipos de desarrollo de software en el contexto de los proyectos y/o la organización.

Gestión Ágil de Proyectos: La respuesta alude a actividades relacionadas con la administración de los proyectos utilizando metodologías ágiles.

Gestión Tradicional de Proyectos: La alternativa de respuesta hace referencia a actividades relacionadas con la administración de los proyectos utilizando metodologías tradicionales, basadas por ejemplo en PMBOK, CMMI o RUP.

Modelado de Arquitecturas de Software: Esta opción de respuesta refiere a actividades relacionadas con el diseño de la arquitectura del software.

Diseño de Interacción Humano-Máquina: La alternativa de respuesta hace referencia a actividades relacionadas con la administración de los proyectos utilizando metodologías ágiles.

Manejo de Herramientas para Testing Automatizado: La opción de respuesta alude a competencias relacionadas con el manejo de herramientas para automatización del testing de software.

Manejo de Herramientas para Gestión de Configuración/Integración Continua: En este caso la opción de respuesta hace referencia a competencias relacionadas con el manejo de herramientas para mantener la integridad del producto de software que se desarrolla.

¿Otras? ¿Cuáles?: Se incluye una pregunta abierta para permitir a los encuestados describir otras competencias técnicas, en caso de no estar incluida en alguna de las opciones anteriormente descriptas.

13. ¿En los equipos de desarrollo de software en los cuáles trabajó o trabaja actualmente, qué competencias conductuales son requeridas? Puede marcar más de una.

Las alternativas posibles fueron:

1. Resolución de problemas complejos
2. Trabajo en Equipo
3. Autoorganización
4. Manejo de Inglés
5. Pensamiento Crítico
6. Inteligencia Emocional
7. Orientación a Servicios
8. Negociación
9. Flexibilidad Cognitiva
10. Otras, ¿cuáles?

Resolución de problemas complejos: Esta opción de respuesta hace referencia a competencias conductuales requeridas para resolver problemas en el contexto de desarrollo de software.

Trabajo en Equipo: Esta opción de respuesta refiere a las competencias conductuales que para facilitar el desarrollo de software en el contexto de los equipos de desarrollo de software son requeridas.

Autoorganización: La opción de respuesta alude a competencias conductuales requeridas para resolver de manera autónoma las actividades necesarias, en el contexto de desarrollo de software.

Manejo de Inglés: Se trata de una opción que indaga sobre competencias relacionadas al manejo del inglés técnico, requerido para trabajar en el contexto de equipos de desarrollo de software.

Pensamiento Crítico: Se buscó conocer, las competencias conductuales vinculadas al desarrollo de pensamiento crítico, que son requeridas para trabajar en el contexto de equipos de desarrollo de software.

Inteligencia Emocional: La opción de respuesta hace referencia a competencias conductuales vinculadas a la inteligencia emocional, necesaria para trabajar en el contexto de equipos de desarrollo de software.

Orientación a Servicios: Se trata de una opción de respuesta que hace referencia a competencias conductuales vinculadas al desarrollo actitudes de orientación a servicio, requeridas para trabajar en el contexto de equipos de desarrollo de software.

Negociación: A través de esta opción de respuesta se busca conocer si las competencias conductuales vinculadas a habilidades de negociación son necesarias y requeridas en el contexto de desarrollo de software.

Flexibilidad Cognitiva: Se trata de una opción de respuesta que hace referencia a competencias conductuales como la flexibilidad cognitiva, requeridas en el contexto de desarrollo de software.

Otras, ¿cuáles?: Se incluyó una pregunta abierta para permitir a los encuestados describir otras competencias conductuales, en caso de no estar incluida en alguna de las opciones anteriormente descritas.

14. ¿Cuál es la formación más adecuada para los integrantes de un equipo de desarrollo de software?

Las opciones que se presentaron en la encuesta fueron:

1. Título Universitario de Grado
2. Título Universitario Intermedio
3. Tecnicatura
4. Estudios no formales
5. Estudiantes de carreras universitarias
6. La formación no afecta en la conformación de los equipos de desarrollo

Título Universitario de Grado: A través de esta respuesta el encuestado que la elige considera que la formación más adecuada para los integrantes de un equipo de desarrollo de software es la que otorga un título universitario de grado.

Título Universitario Intermedio: Con esta opción de respuesta el encuestado considera que la formación más adecuada para los integrantes de un equipo de desarrollo de software es la que otorga un título universitario intermedio. Las carreras de grado otorgan frecuentemente a los dos años un título de *Técnico en Programación* y a los 3,5 o 4 años un título de *Analista Universitario de Sistemas*, estas titulaciones, son de relevancia, dado que mayoría de los estudiantes se incorporan a la vida laboral durante los primeros años de la carrera, y muchos de ellos prolongan sus estudios o nunca los finalizan y estos títulos intermedios, les dan un reconocimiento académico.

Tecnicatura: con esta opción de respuesta el encuestado que la elige considera que la formación más adecuada para los integrantes de un equipo de desarrollo de software es la que otorga una tecnicatura.

Estudios no formales: La elección de esta respuesta implica que el encuestado que la elige considera que la formación más adecuada para los integrantes de un equipo de desarrollo de software es la que se obtiene por medio de estudios no formales.

Estudiantes de carreras universitarias: A través de la elección de esta opción el encuestado que la elige considera que la formación más adecuada para los integrantes de un equipo de desarrollo de software es la de estudiantes de carreras universitarias de informática.

La formación no afecta en la conformación de los equipos de desarrollo: Esta alternativa de respuesta implica que el encuestado que la elige considera que la educación formal no afecta en la conformación de equipos de desarrollo de software.

15.- Según su criterio ¿Cuáles son los aspectos que conspiran contra la obtención de un producto de software de calidad?

En este ítem las alternativas de respuesta fueron:

1. Falta de gestión
2. Falta de comunicación
3. Falta de liderazgo
4. Falta de conocimiento técnico
5. Escaso o nulo nivel de involucramiento del Cliente
6. Malas relaciones interpersonales
7. Malas estimaciones
8. Compromisos asumidos con poca información
9. Subestimación del producto a construir
10. Mala o escasa adaptación de la metodología o proceso a utilizar
11. Otras, ¿cuáles?

Falta de gestión: Esta opción de respuesta indica que el encuestado considera que la inexistencia y/o falta de gestión de los proyectos conspira con la obtención de un producto de software de calidad.

Falta de comunicación: En este caso, la respuesta del encuestado estaría indicando que considera que la inexistencia y/o falta de comunicación en el contexto de los proyectos conspira contra la obtención de un producto de software de calidad.

Falta de liderazgo: Si el encuestado ha optado por esta respuesta es que considera que la inexistencia y/o falta de liderazgo en los proyectos conspira contra la obtención de un producto de software de calidad.

Falta de conocimiento técnico: Esta respuesta está indicando que el encuestado considera que la falta de conocimiento técnico conspira contra la obtención de un producto de software de calidad.

Escaso o nulo nivel de involucramiento del Cliente: En este caso, si el encuestado ha realizado esta opción es porque considera que la inexistencia y/o falta de gestión de los proyectos conspira con la obtención de un producto de software de calidad.

Malas relaciones interpersonales: La opción de respuesta indica que el encuestado considera que la existencia de malas relaciones interpersonales entre los involucrados en un proyecto conspira contra la obtención de un producto de software de calidad.

Malas estimaciones: Se trata de una opción de respuesta que indica que la realización de malas estimaciones conspira contra la obtención de un producto de software de calidad.

Compromisos asumidos con poca información: En este caso, la respuesta indica que el encuestado considera que la existencia de compromisos asumidos con poca información en el contexto de un proyecto conspira contra la obtención de un producto de software de calidad.

Subestimación del producto a construir: La respuesta señala que el encuestado considera que la acción de simplificación excesiva de lo necesario para construir el producto de software conspira contra la obtención de un producto de software de calidad.

Mala o escasa adaptación de la metodología o proceso a utilizar: Se trata de una opción que pone el énfasis en que la escasa o inexistente adaptación de la metodología a proceso a utilizar para el desarrollo del producto de software conspira contra la obtención de un producto de software de calidad.

¿Otras? ¿Cuáles?: Se incluyó una pregunta abierta para permitir a los encuestados describir la metodología utilizada en caso de no estar incluida en alguna de las opciones anteriormente descritas.

16.- Según su criterio, ¿cuál de los siguientes aspectos de la relación vincular entre los involucrados en el desarrollo de un producto de software, es necesario mejorar?

Las alternativas de respuesta fueron:

1. Comunicación
2. Compromiso
3. Involucramiento
4. Cooperación
5. Otro, cuál

Comunicación: Se refiere a la acción consciente de intercambiar información entre dos o más miembros del equipo de desarrollo, con el fin de transmitir o recibir significados a través de un sistema compartido de signos y normas semánticas.

Compromiso: En este caso, se alude a la capacidad que tiene una persona para tomar consciencia de la importancia que existe en cumplir con algo acordado anteriormente. La selección de esta respuesta implica que el encuestado considera que ser una persona que cumple con sus compromisos es considerado un valor y una virtud, ya que esto suele asegurar el éxito en los proyectos.

Involucramiento: Se refiere al compromiso activo para prestar atención, cumplir con trabajos y encontrar algún valor inherente en lo que se realiza. Es decir que para el encuestado es importante que los miembros del equipo relacionen las tareas asignadas con sus metas a corto y largo plazo.

Cooperación: Se trata de una opción que se refiere a la elección por parte de los miembros del equipo de desarrollo, de roles diferenciados y complementarios para el logro de la tarea sin que esto signifique reemplazar ni desplazar a ningún otro miembro.

¿Otro? ¿Cuál?: La inclusión de una pregunta abierta responde al interés de permitir a los encuestados describir otro aspecto de la relación vincular, en caso de no estar incluido en alguna de las opciones anteriormente descritas.

17.- ¿Qué recomendaciones haría usted para la formación de un equipo de desarrollo de software eficiente?

Esta es una pregunta abierta, de opinión para los encuestados con la intención de que compartan su experiencia y obtener con ello una validación a la hipótesis del trabajo de investigación.

Instrumento conceptual para la Encuesta : Sistema de Matrices de Datos

Se adopta la propuesta de Samaja (2010) quien sostiene que la estructura general del dato científico tiene cuatro componentes: la unidad de análisis (UA), las variables (V), los valores (R) y los indicadores (I).¹⁶.

Sobre la base del instrumento desarrollado, se procedió a realizar el pilotaje a 10 personas, a fin de evaluar la consistencia interna del mismo para dotar al trabajo de validez y confiabilidad. Posteriormente se realizó la medición de alfa de Cronbach , que dio 0,48, que si bien no resulta alto (cercano a 1), es aceptable para este tipo de estudios que miden percepciones, comportamientos y requerimientos, como es el caso de la presente investigación. A partir de este resultado, se procedió a ajustar la encuesta que finalmente se administró. (Ver Anexo I).

¹⁶ El autor sostiene que en toda investigación científica hay más de una matriz de datos, es por esto que sugiere que es más correcto considerar la noción de "sistema de matrices de datos"¹⁶. Como mínimo tres matrices de datos: la matriz de anclaje o matriz focal, es la matriz en la que se ha decidido anclar la investigación denotándola N_a ; la matriz subunitaria constituida por las componentes de la matriz focal, que se denota N_{-1} y la matriz supraunitaria o matriz de los contextos de la matriz focal, que se denota N_{+1} .

Cuadro 1. Matriz del nivel Supra Unitario

Unidad de Análisis	Variables	Dimensiones	Indicadores	Escalas	Valores
Equipos de trabajo	Dinámica de trabajo	Localización geográfica del equipo de trabajo	Nivel de dispersión que los integrantes del equipo tienen en relación con su asiento geográfico	Nominal	Co-locado Distribuido Mixto u otro
	Metodologías de trabajo	Tipo de metodologías	Metodologías basadas en procesos definidos	Nominal	PUD (Proceso Unificado de Desarrollo) RUP (Rational Unified Process) ICONEX, Yourdon, Microsoft Solution Framework, No utilizo metodologías basadas en procesos definidos, otras l
			Metodología Frameworks basados en filosofía lean Ágil,	nominal	Scrum, XP, FDD, kanban, Srumban, ATDD/TDD, no utiliza metodologías Lean/Agiles
			Metodologías de gestión de proyectos	nominal	Basada en PMI, CMMI, Gestión Ágil, lean, RUP, No utiliza, otras

Cuadro 2. Matriz del Nivel de Anclaje

Unidad de Análisis	Variables	Dimensiones	Indicadores	Escalas	Valores
Integrantes de los grupos de trabajo	Características personales	Sexo	Sexo de cada uno de los integrantes	Nominal	Femenino Masculino
		Edad	Edad en años de cada uno de los integrantes	De Razón	Menos de 26 años Entre 26 y 32 Entre 33 y 39 Entre 40 y 45 Más de 46
	Características laborales	Titulación	Tipo de titulación	Nominal	Ingeniero en sistemas Informáticos Licenciado en Informática o similar
		Actividad laboral	Tipo de actividad que desempeña	Nominal	Desarrollo de Software Scrum master Supervisor de Aplicaciones Líder de proyecto Desarrollador SR
		Asiento de la actividad laboral	Localidad en la que trabaja	Nominal	Córdoba Buenos Aires Otras
		Antigüedad	Antigüedad en el trabajo	De intervalo	{1,2,3,4,5,6,7.....}
		Tipo de tareas que desarrolla en el trabajo	Principales actividades que desarrolla en el trabajo	Nominal	Gestión de Proyecto Arquitectura Ingeniería de Requerimientos Despliegue Diseño Implementación Otras

Cuadro 3. Matriz del nivel Subunitario

Unidad de Análisis	Variables	Dimensiones	Indicadores	Escala	Valores
Atributos personales como profesionales del sector	competencias	competencias técnicas	Requerimientos de la competencia por el sector empleador	Nominal	Programación o Desarrollo Manejo de Técnicas de Ingeniería de Requerimientos (Casos de Uso, Técnicas de Elicitación) Gestión ágil de requerimientos (user stories) Testing de Software Gestión de Personas Gestión Ágil de Proyectos Gestión Tradicional de Proyectos Modelado de Arquitecturas de Software Diseño de Interacción Humano-Máquina Manejo de Herramientas para Testing Automatizado Manejo de Herramientas para Gestión de Configuración/Integración Continua
		competencias conductuales	Requerimiento de la competencia por el sector empleador	Nominal	Resolución de problemas complejos, Trabajo en Equipo, Autoorganización, Manejo de Inglés, Pensamiento Crítico, Orientación a Servicios, Negociación
		Ausencia de competencias técnicas o conductuales	Aspectos que conspiran contra la obtención de un producto software de calidad	Nominal	Falta de gestión, Falta de comunicación, Falta de liderazgo, Falta de conocimiento técnico, Escaso o nulo nivel de involucramiento del Cliente, Malas relaciones interpersonales, Malas estimaciones, Compromisos asumidos con poca información, Subestimación del producto a construir, Mala o escasa adaptación de la metodología o proceso a utilizar
	Relación vincular		Principales aspectos de la relación vincular entre los involucrados en el desarrollo de un producto de software		Comunicación, Involucramiento Compromiso Cooperación Otros
	Formación		Nivel de formación requerido	Ordinal	Título universitario de grado Título universitario de posgrado

Focus Group

El grupo focal es una herramienta muy útil para la planificación de los programas y la evaluación de estos. El secreto consiste en que los participantes puedan expresar libremente su opinión sobre diferentes aspectos de interés en un ambiente abierto para el libre intercambio de ideas. Otro de los aspectos positivos estriba en el hecho de proveer participación a las personas involucradas en los respectivos programas. La actividad la puede dirigir cualquier persona que sea adiestrada y adquiera las destrezas requeridas, y tenga un interés genuino en llevar a cabo la mencionada dinámica grupal.

Características de los grupos focales

- Los participantes tienen ciertas características homogéneas.
- Se proveen datos de índole cualitativo.
- La discusión es enfocada en un aspecto específico.
- Es sumamente importante la percepción de los usuarios y consumidores sobre productos, servicios y oportunidades.
- El propósito no es el de establecer consenso, sino el de establecer las percepciones, sentimientos, opiniones y pensamientos de los usuarios sobre productos, servicios y oportunidades.
- Puede constituir un buen foro para facilitar un cambio sistémico adecuado en la organización.
- Es un proceso adecuado para facilitar el aprendizaje de los miembros de la organización.

En relación con la constitución de los grupos, es importante para conformar un focus group determinar la población a participar. Estos pueden ser los miembros de grupos específicos, consejos asesores, empleados, consumidores o clientes de algún producto o programa específico. Se espera que los participantes en un mismo grupo sean lo más homogéneos posibles y no se conozcan entre sí. El propósito es que los participantes se puedan expresar libremente sin herir susceptibilidades. Sin embargo, aun cuando los participantes se conozcan, es posible desarrollar buenos grupos focales si el moderador provee un ambiente apropiado para el mismo. El tamaño de cada grupo debe fluctuar entre 4-10 personas, con un número óptimo de 6-8.

En el presente estudio, se seleccionó un grupo de 6 profesionales (mujeres) entre 24 y 59 años, quienes participaron libremente de un dialogo abierto coordinado por el investigador sobre la base de preguntas prediseñadas, que se fueron incorporando paulatinamente durante la conversación, con el objetivo específico de relevar aspectos vinculados al género en el desempeño profesional relacionado con el desarrollo de software.

CAPITULO VI

Resultados

Análisis Descriptivo

Caracterización general de la muestra

a) Edad de los encuestados

Al analizar la edad de los encuestados, se observa que aunque la correspondencia no es estricta, los grupos etarios pueden asimilarse a la segmentación del marketing digital, trabajada por Mabel Cajal, la que se corresponde a su afinidad con la tecnología, más específicamente con las redes sociales y el turismo 2.0 o turismo electrónico. (ver Gráfico 13).

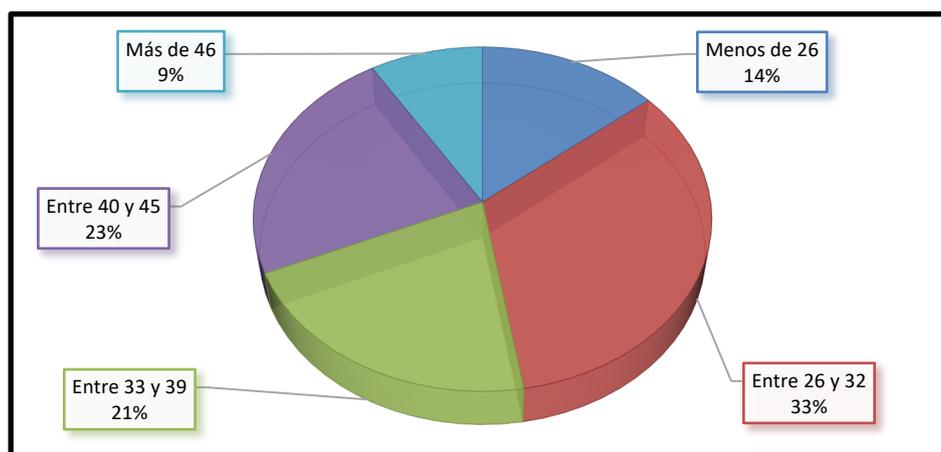


Gráfico 13. Edad de los encuestados
Fuente: Elaboración propia

En este sentido se habla de:

- Generación Z aquellos que tienen menos de 26 años
- Millennials quienes tienen entre 26 y 32 años
- Generación X , con edades que oscilan entre 33 y 39 años
- Baby Boomers, incorpora en esta categoría a quienes tienen más de 46 años.

Mabel Cajal (Cajal, 2018) define a las generaciones según su comportamiento on line, de la siguiente forma:

Baby Boomers (1946-1964): Reciben este nombre por haber nacido en una época de incremento en las tasas de natalidad. Viven un cambio en el modelo tradicional con el comienzo de la incorporación de la mujer en el mercado laboral. El trabajo es de gran importancia, son comprometidos, se observa que son fieles a su vida profesional y con muy buenas condiciones para crear equipos. Se trata de un segmento acostumbrado a trabajar muchos años de su vida en la misma empresa, sin embargo ven como mentores a los denominados millennials ya que les dan ideas creativas.

Generación X (1965-1978): En este caso, se observa como la mujer se incorpora en un mayor número y de manera más habitual al mercado laboral por la necesidad de cubrir los gastos diarios, aunque todavía existe un alto porcentaje que se ocupa de la gestión de la casa. Son personas que saben trabajar en equipo y fomenta las relaciones laborales de confianza. Se trata de sujetos estables quienes en el ámbito laboral obtienen buenos resultados. Se caracterizan además porque desean el desarrollo profesional dentro de la misma empresa y reflexionan antes de dar un cambio a sus vidas profesionales. Un aspecto central es que vivieron la llegada de internet a sus vidas así como todo el desarrollo tecnológico posterior, esta circunstancia podría influir en que es la generación más adaptable al cambio.

Millennials o Generación Y(1979-1996): En este caso, se observa que la mujer no duda en desarrollar su etapa profesional, y en general no están dispuestas a renunciar a ella. Asimismo, se advierte, tanto en hombres como en mujeres una marcada predisposición a mantener su estado de solteras/os o casados sin hijos. Por otro lado, se trata de un segmento emprendedor al que le atrae aprender, así como formar parte de la toma de decisiones y ganar reconocimiento. Otro rasgo que los caracteriza es que no conciben la vida sin tecnología y fundamentalmente se adaptan al ritmo cambiante de las cosas, muestran alta tolerancia al fracaso. Finalmente otra característica que los distingue es su gusto por los viajes.

Generación Z (Desde 1995): Son jóvenes recientemente incorporados al mercado laboral, que en su gran mayoría desea tener su propia empresa y que sus hobbies se conviertan en fuente de ingresos. Buscan trabajos flexibles, a distancia y que les permita desempeñarse en varias actividades a la vez. En este caso, decididamente su vida gira en torno a la tecnología, por ejemplo estudian y leen on line, recurren a YouTube para informarse. Se trata de sujetos a quienes resulta difícil captar su atención, son creativos y muestran una significativa preocupación por el medio ambiente y por las relaciones personales.

Los datos obtenidos, indican que el 54 % de nuestra población se encuentra entre los 26 y 39 años, lo que, como se señaló, resulta coincidente, con la tipología descrita, en función de que la población encuestada, son profesionales del área informática (Ver Gráfico 13).

b) Actividad Laboral que informan los encuestados

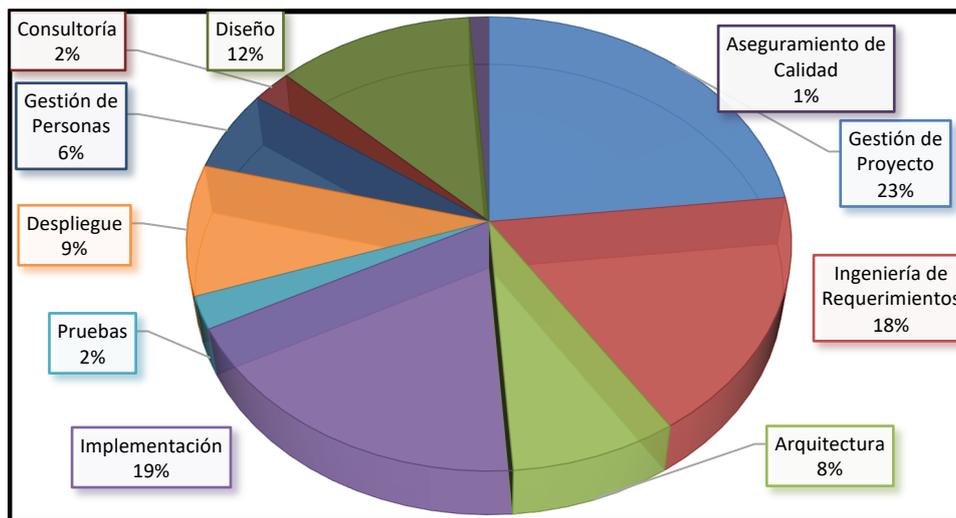


Gráfico 14. Actividad laboral
Fuente: Elaboración Propia

Si se observan los resultados del Gráfico 14, se visualiza la distribución de actividades de los encuestados está vinculada al seniority, considerando esta categoría como la habilidad de saber-estar a la altura en el entorno laboral o experiencia. Este seniority se pone de manifiesto en la cantidad de años de trabajo profesional de los encuestados. Los resultados muestran que el 32% de los encuestados informa que su actividad laboral está vinculada a gestión y/o consultoría y el 38 % de los encuestados informa que trabaja en actividades de perfil técnico, las que también requieren más años de experiencia para realizarlas, como por ejemplo arquitectura, diseño e ingeniería de requerimientos que son actividades laborales cuyo seniority se construye iniciando primero en otras como la implementación, las pruebas y el despliegue y luego evolucionan hacia los perfiles mencionados.

Por su parte, las actividades de implementación (programación), prueba y despliegue (instalación y puesta en producción) representan al 30% de los encuestados. Actividades por las que se inicia más comúnmente en el desarrollo profesional.

c) Sexo de los encuestados

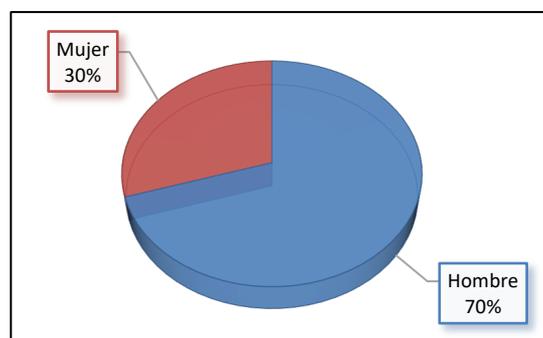


Gráfico 15. Sexo
Fuente: Elaboración propia

De la lectura del Gráfico 15, se ve claramente un importante porcentaje de hombres (70%) sobre el de mujeres (30%) . Estos resultados son un reflejo de la situación actual en la profesión informática.

Dado el resultado de este punto, se decidió realizar un focus group, con un conjunto de 6 profesionales, las que ya habían sido relevadas en la encuesta. De esta actividad que duró 3 horas¹⁷, se obtuvieron los resultados que se describen a continuación:

Las profesionales con más experiencia laboral y profesional plantearon que en la década de los noventa el porcentaje de mujeres era superior al de los hombres. Esta tendencia fue disminuyendo con el tiempo. Al consultarlas sobre el motivo, manifestaron que dado que el mercado tiene una demanda creciente, exponencial de acceso a software, las empresas que trabajan proveyéndolo incrementaron su nivel de exigencia para con sus profesionales, exigencias puestas de manifiesto en la cantidad de horas diarias de trabajo, tomando también días no laborables. Otra exigencia está vinculada a la demanda de viajes laborales, no reemplazables con el teletrabajo o el homework.

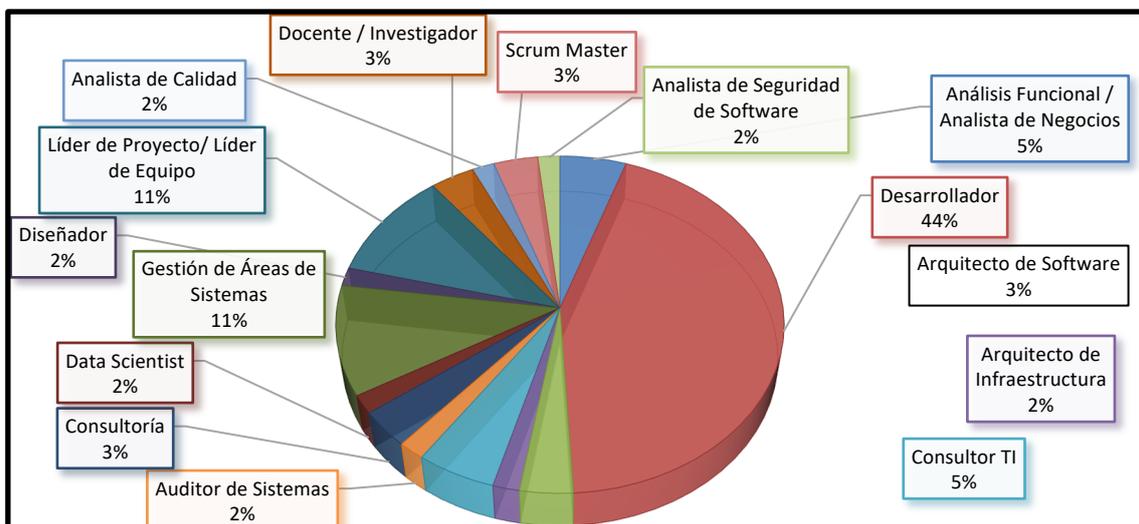


Gráfico 16. Inserción en el ámbito laboral
Fuente: Elaboración propia

Al observar los resultados que se presentan en el Gráfico 16, se advierte que el 44 % corresponde a la actividad de desarrollador, en tanto que Líder de proyecto/Equipo y Gestión de áreas de sistemas se ubican en un 11% para cada una de las actividades. Para el resto de las tareas, los resultados oscilan entre el 2 al 3% con excepción de Análisis Funcional/ Analista de negocios y Consultor TI que ambas obtuvieron un 5 % de las respuestas. La dispersión que se advierte está basada en el tipo de muestra elegida, muchos de ellos desarrollan actividad académica, por ser esta población cercana al tesista. Cabe agregar que por tener una importante formación académica y profesional, la mayoría de ellos pueden desempeñarse en diversas tareas, desempeñándose muchos de ellos como consultores y por utilizar modalidades y

¹⁷ En el capítulo V se describe la metodología utilizada

metodologías de trabajo, donde como miembros de equipos de desarrollo desempeñan diversos roles dentro de los proyectos.

d) Formación o Título

En relación con esta variable, el 93% cuenta con título universitario, distribuidos de la siguiente manera 84% títulos de grado en tanto que el 9% títulos de pregrado. (ver Gráfico 17).

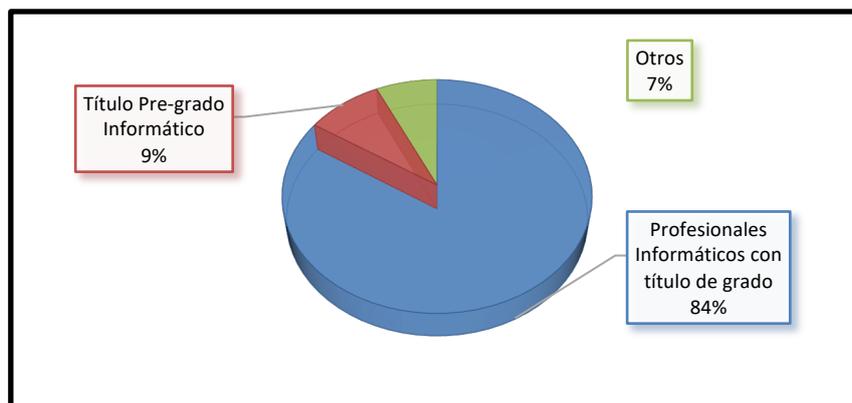


Gráfico 17. Formación o título
Fuente: Elaboración propia

e) Localidad donde trabaja

En relación con el lugar de radicación del empleo, se advierte una fuerte concentración en Córdoba (47%) y Ciudad Autónoma de Buenos Aires (46%) (Ver Gráfico 18). Cabe aclarar que ambas son consideradas localidades maduras para la industria de software¹⁸. Se consideran localidades maduras aquellas que albergan a empresas consolidadas en la actividad, fundamentalmente apoyadas por una masa crítica de recursos humanos formados en ámbitos universitarios.

¹⁸ Argentina. Ministerio de la Producción de la Nación. Programa de Desarrollo Regional de Empresas y clústeres más competitivos

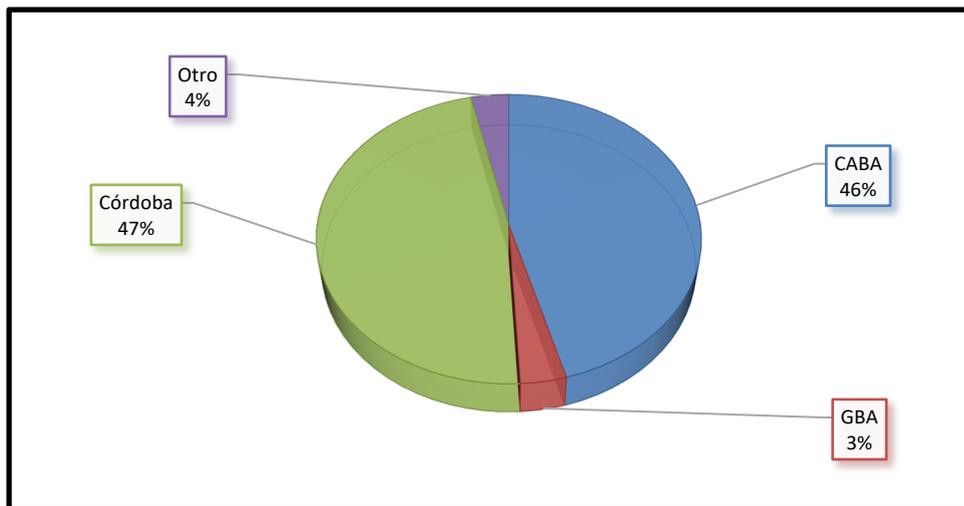


Gráfico 18. Lugar de radicación laboral
Fuente: Elaboración propia

f) Antigüedad en la última función

A pesar de que el grupo entrevistado es de un grupo etario de edad media para la profesión y más de un 90% son profesionales, o sea tienen un seniority alto, este gráfico de antigüedad denota una importante rotación laboral. A partir de estos resultados podría inferirse como ventaja la posibilidad del crecimiento vegetativo dentro de las organizaciones y como desventaja podemos mencionar la falta de pertenencia (Ver Gráfico 19)

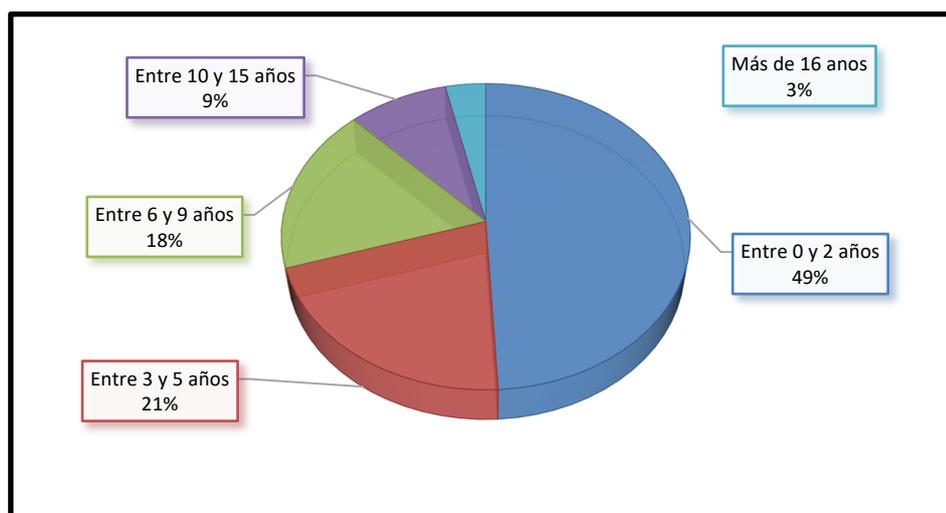


Gráfico 19. Antigüedad en la última función
Fuente: Elaboración propia

Modalidades y Metodologías de Trabajo

a) Forma de trabajo Distribuida o Co-locada

En relación con la modalidad de trabajo, los resultados muestran que el 40 % trabajan bajo la modalidad distribuida, en tanto que el 51% en la co-locada. Solo un 9% lo hacen en la modalidad mixta. Se recuerda, que en el primer caso es cuando el equipo trabaja distribuido geográficamente. Esta posibilidad se da por la mediación tecnológica a la que acuden los equipos. En tanto que los equipos co-locados son aquellos que trabajan físicamente en el mismo ambiente. (ver Gráfico 20)

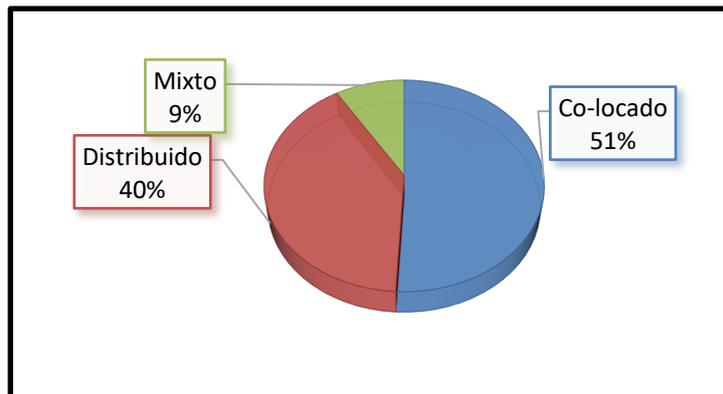


Gráfico 20. Forma de Trabajo
Fuente: Elaboración Propia

b) Metodologías de trabajo utilizadas

Al analizar las respuestas vinculadas a las metodologías utilizadas se obtuvieron los resultados que se presentan a continuación.

En relación con los procesos definidos (Ver Gráfico 21), se observa que mayoritariamente no son utilizados, ya que el 54 % respondió que no los usa. El 15% utiliza métodos PUD y el 10% RUP, en tanto que solo el 8% utiliza metodologías Ágiles.

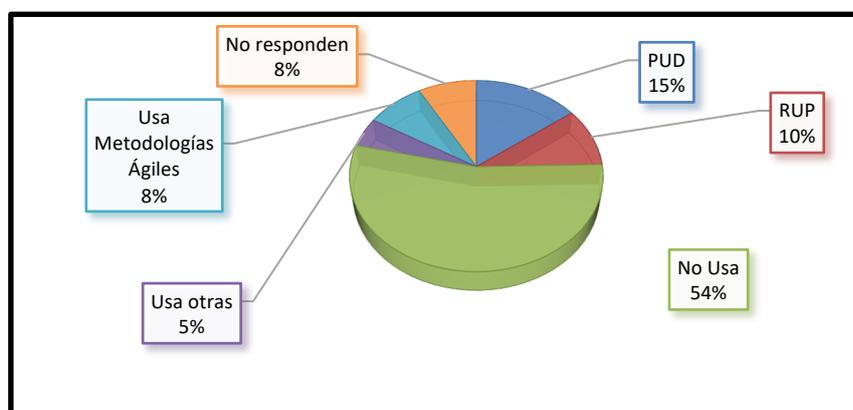


Gráfico 21. Utilización de Procesos Definidos
Fuente: Elaboración propia

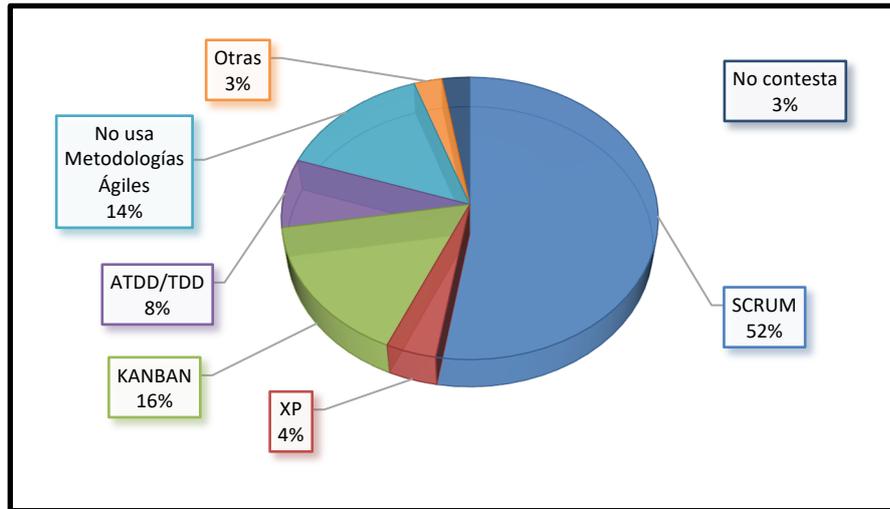


Gráfico 22. Metodologías basadas en procesos empíricos
Fuente: Elaboración propia

Por otra parte, al indagar respecto de las metodologías basadas en procesos empíricos, se observa al igual que en el caso anterior una tendencia que está dada por el uso de SCRUM (52%) de las respuestas confirmó su utilización. Muy por debajo de ese porcentaje encontramos KANBAN (16%) y ATDD/TDD (8%). Por su parte el 14% de los encuestados señaló que no utiliza metodologías Ágiles (Ver Gráfico 22)

Por otra parte, al consultar sobre la metodología de gestión de Proyectos utilizadas (Gráfico 23), se observa, a diferencia de los casos anteriores, que la más utilizada es la metodología de gestión ágil (56%), seguida aunque con una diferencia significativa por la metodología de gestión basada en PMI con 18%. El resto de las metodologías no alcanzó a obtener más de un 10 %, ya que la basada en RUP obtuvo un 7%, propio o híbrido 5%, y lean 3%.

Por otra parte un 11 % respondió que no utiliza metodologías de gestión de proyectos.

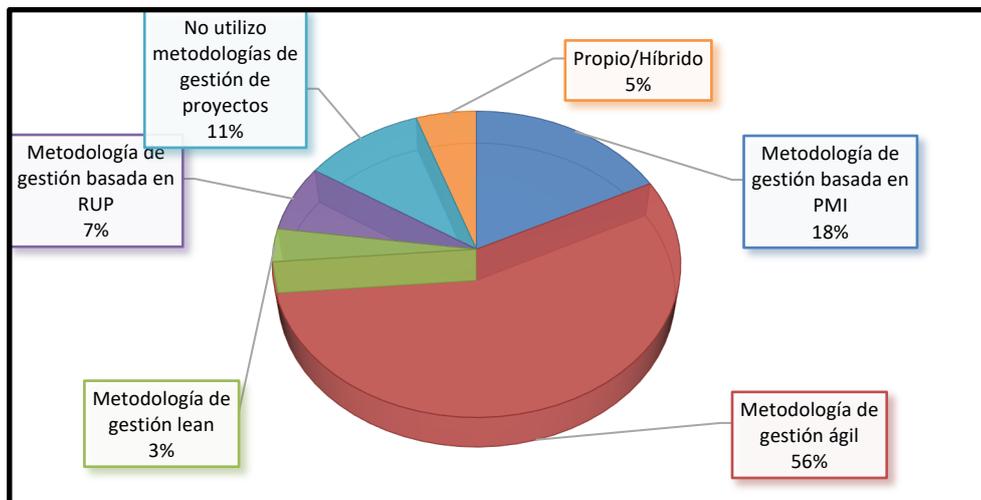


Gráfico 23. Metodología de Gestión de Proyectos utilizadas

Fuente: Elaboración propia

Competencias y Formación requerida para integrar equipos eficientes

a) Competencias Técnicas

Un aspecto que resultaba de gran significancia para nuestro trabajo era indagar cuales eran las competencias técnicas requeridas para desempeñar las distintas posiciones en los ámbitos laborales.

En este sentido, la competencia técnica con mayor requerimiento, de acuerdo con los datos obtenidos, fue la programación o desarrollo con 29%. El resto de las competencias se colocó en un rango muy inferior, es decir entre 10-13%. Es así como encontramos con un 13% testing de software, con un 11 % tanto la Gestión Ágil de requerimientos como la Gestión de personas y un 10 % para modelado de arquitecturas de software. El resto de las competencias técnicas se ubicó por debajo de 10 %: Gestión Ágil de proyectos, 8%, Gestión tradicional de proyectos 7%, Ingeniería de requerimientos 6% y Herramientas de Gestión de Configuración /Integración continua con un 5%. (Ver Gráfico 24)

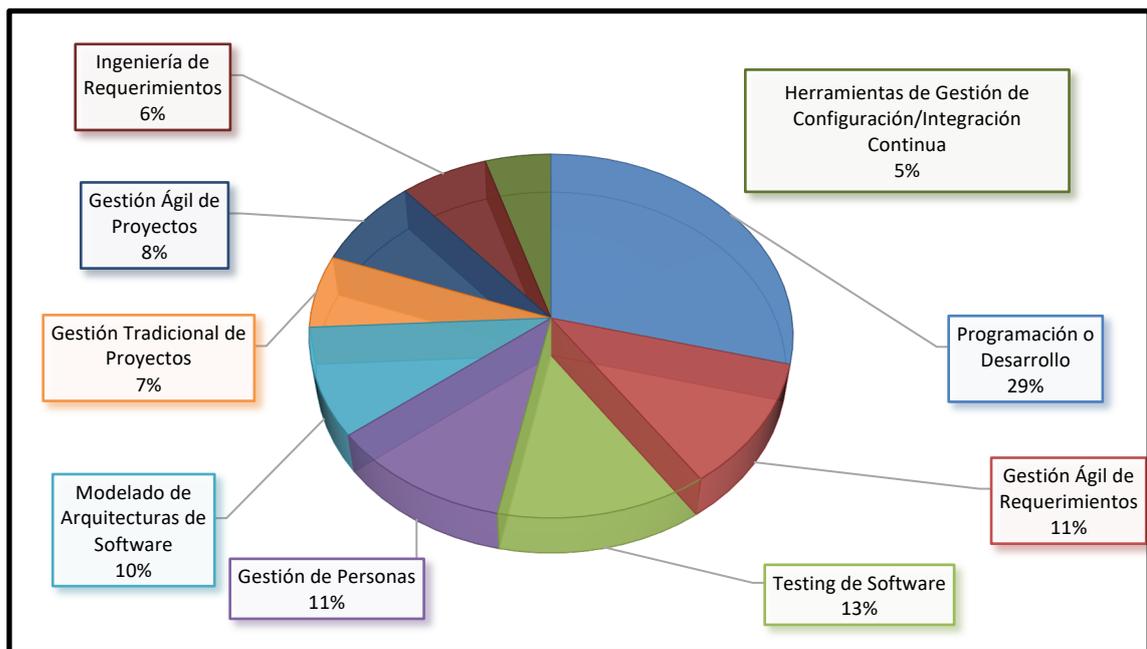


Gráfico 24. Competencias Técnicas requeridas
Fuente: Elaboración propia

b) Competencias Conductuales

Si se observa el Gráfico 25, se advierte que las competencias con mayor requerimiento en el sector son Trabajo en equipo (32%), Autoorganización 27% y resolución de problemas complejos (24%). Entre ellas acumulan el 83% de las demandas conductuales. El resto de las competencias requeridas se

ubican muy lejos en términos porcentuales ya que manejo de inglés y pensamiento crítico obtuvo un 6%, orientación a servicios 4% y flexibilidad cognitiva tan solo un 1%.

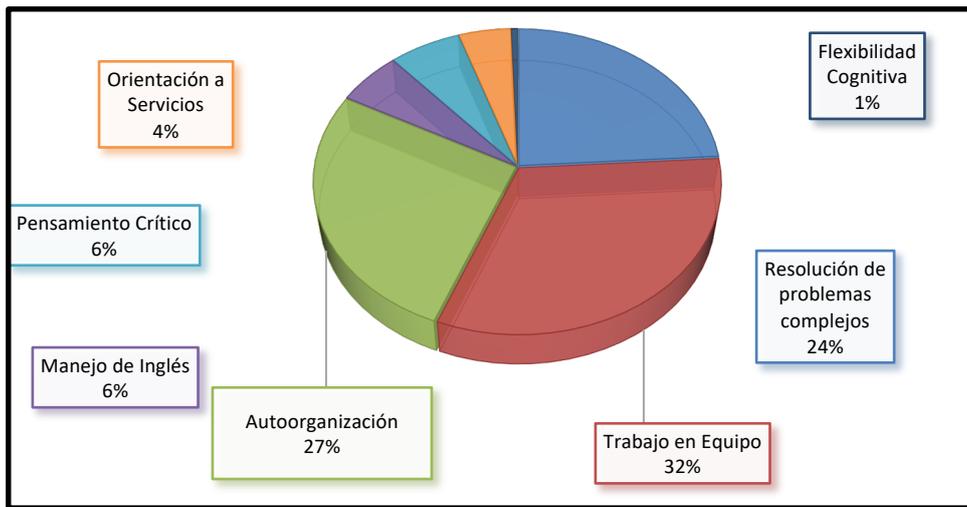


Gráfico 25. Competencias conductuales Requeridas
Fuente: Elaboración propia

c) Formación más adecuada de los equipos de trabajo

Al ser consultados respecto de cual consideraban que era la formación más adecuada que un sujeto tenía que contar para ser integrado a un equipo de trabajo, sin lugar a duda la respuesta fue título universitario de grado (58%), seguido por la opción estudiantes de carreras universitarias (26%). Es decir que el 84% de los encuestados consideró que la universidad a través de sus carreras de grado era la mejor opción para el sector, aun cuando el profesional se encontrara en proceso de formación, tal como es el caso de los estudiantes. Las opciones estudios no formales (5%), tecnicatura (7%) y otras (4%), se ubicaron muy lejos de aquellos valores. (Ver Gráfico 26)

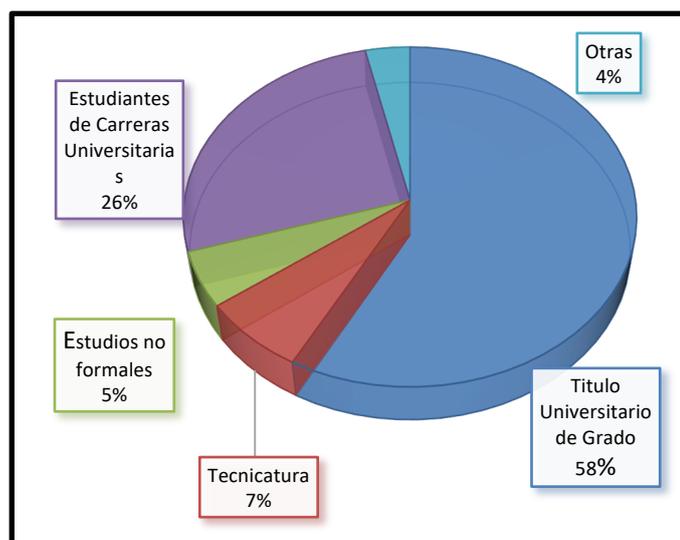


Gráfico 26. Formación más adecuada para integrar equipos de trabajo
Fuente: Elaboración propia

d) Aspectos que conspiran para la obtención de un producto de software de calidad

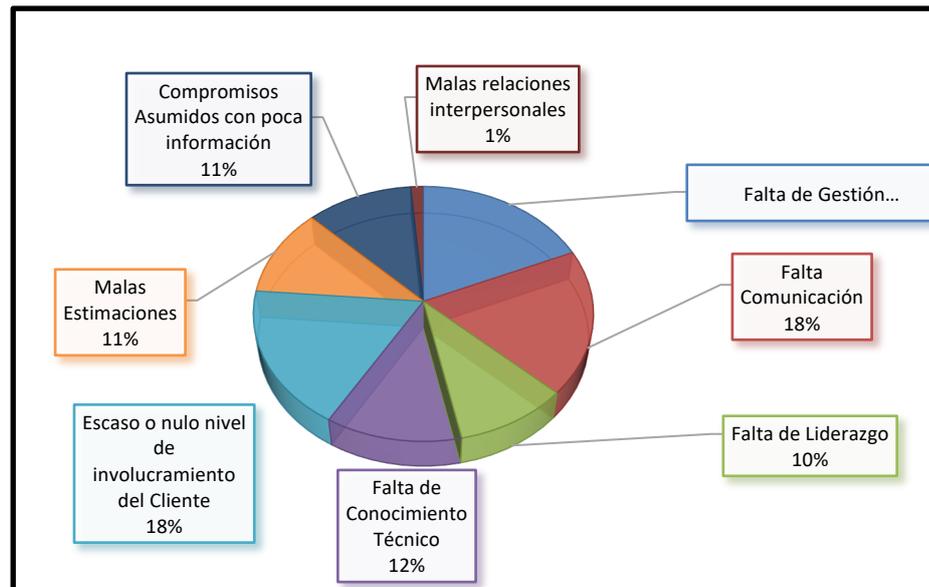


Gráfico 27. Factores que dificultan la producción de software de calidad
Fuente: Elaboración propia

Al ser consultados sobre aquellos factores que contribuían a la producción de software de calidad, (Ver Gráfico 27) el principal aspecto que se señala es la falta de gestión (19%). Es decir que todos aquellos atributos personales que implican una gestión hacia la excelencia serían fundamentales para el proceso. Por otro lado y muy cerca de dicha ponderación aparecen, los déficits de involucramiento del cliente por un lado y la falta de comunicación por otro, ambos con un 18 %. Por otra parte y con un peso algo menor encontramos la falta de conocimiento técnico (12%) y con una valoración cercana, aparecen las malas estimaciones y los compromisos asumidos con poca información, ambas con 11 % de respuestas. Con una valoración un poco menor, surge como factor interviniente la falta de liderazgo (10%).

e) Aspectos de la relación vincular entre los involucrados en el desarrollo de un producto de software que es necesario mejorar

Al observar los datos que surgen del Gráfico 28, encontramos que la comunicación (36%) es el aspecto de la relación vincular que aparece como central para el desarrollo de software de calidad. Dato que coincide con los resultados obtenidos en el apartado anterior. Estos resultados nos permiten inferir la importancia de la comunicación en los procesos que este trabajo indaga.

Por otro lado, la cooperación que tal como se señaló, se refiere a la elección por parte de los miembros del equipo de desarrollo, de roles diferenciados y complementarios para el logro de la tarea sin que esto signifique reemplazar ni desplazar a ningún otro miembro. Este aspecto, en nuestro estudio fue ponderado con un 27%.

El compromiso entendido como capacidad de una persona para tomar consciencia de la importancia que existe en cumplir con algo acordado anteriormente y el involucramiento, que como especie de compromiso, en el sentido de habilidad para prestar atención, cumplir con trabajos y encontrar algún valor inherente en lo que se realiza obtuvieron el 18 % y 17 % respectivamente.

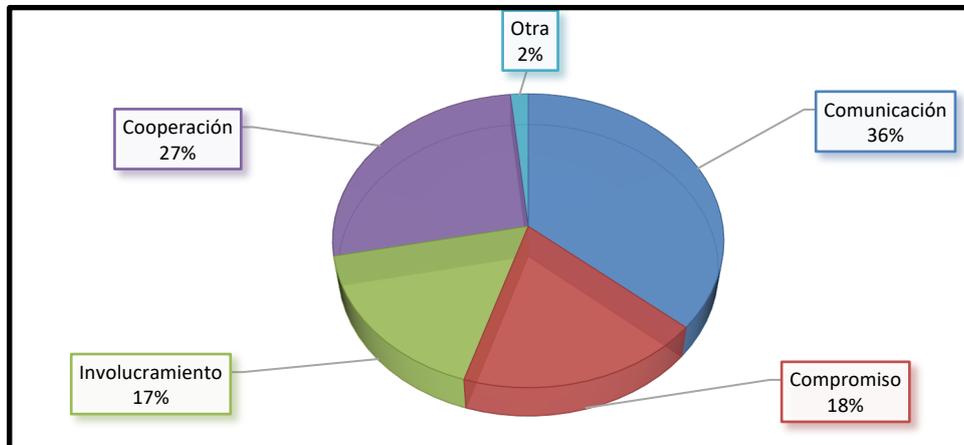


Gráfico 28. Relaciones vinculares significativas para desarrollar software de calidad
Fuente: Elaboración propia

Análisis Bivariado

A continuación se presentan los resultados del análisis bivariado, obtenidos a partir de los datos de la encuesta. En todos los casos se trabajó con un nivel de significación del 5% ($\alpha=0,05$). Todas las Tablas de Frecuencias se presentan como Anexo II

Resultados para las variables Edad y Antigüedad en la última función de trabajo

Los valores previstos para la variable Edad fueron , menos de 26 años, entre 26 y 32 años, entre 33 y 39 años , entre 40 y 45 años y más de 46. La escala fue establecida con esos intervalos, ya que el conocimiento previo que se tiene de la movilidad laboral indica que son esas las edades cruciales en el mercado laboral del sector informático. Para la variable antigüedad, los valores se dejaron abiertos, por lo cual, a partir de las respuestas obtenidos, se procedió a agrupar y a categorizar las repuestas. Los resultados obtenidos pueden observarse en el Gráfico 29.

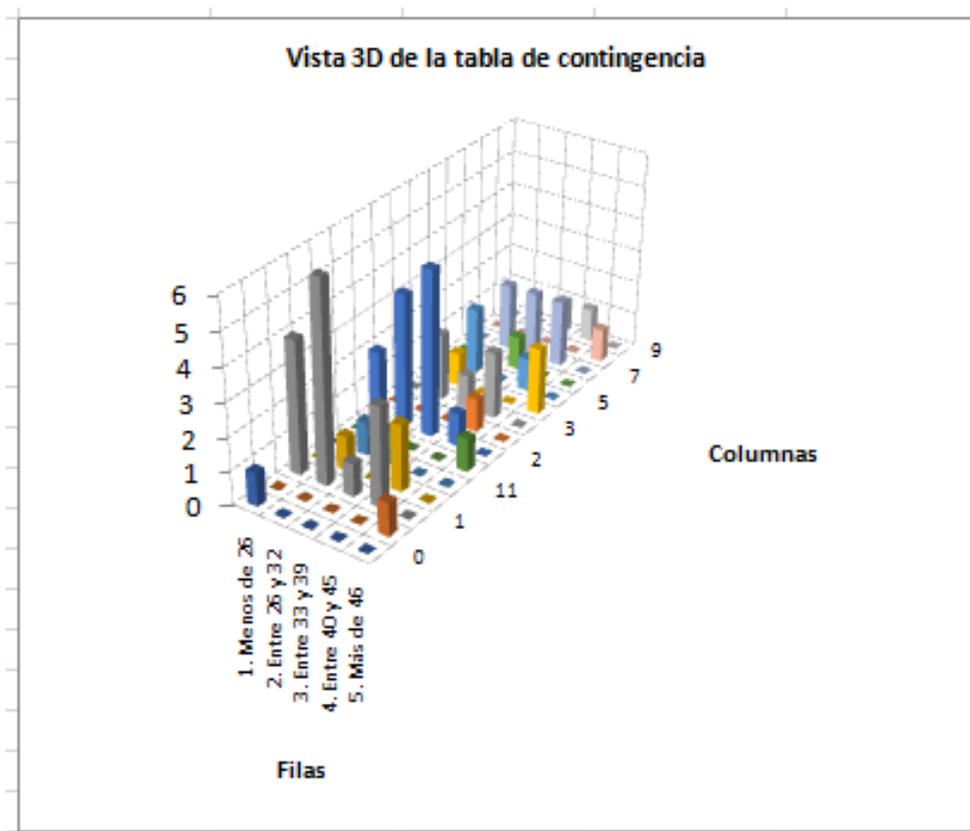


Gráfico 29- Tabla de contingencia para las variables Edad y Antigüedad en la última función en el trabajo
Fuente: Elaboración Propia

Tabla 1. Prueba de chi cuadrado de Independencia para variables edad y Antigüedad en la última función laboral
Fuente: Elaboración Propia

Chi-cuadrado ajustado (Valor observado)	78,740
Chi-cuadrado ajustado (Valor crítico)	74,468
p-valor	0,024

Los resultados de la Tabla 1 permiten afirmar que existe dependencia entre las filas y columnas de la tabla de contingencia, ya que el p-valor computado (0,024) es menor que el nivel de significancia $\alpha=0,05$, por lo tanto se rechaza la hipótesis nula H_0 y se acepta la hipótesis alternativa H_a .

Esto nos permite señalar que, con los datos de la experiencia realizada existe dependencia entre la edad y los años de antigüedad en el último trabajo. Estos resultados serían indiciarios de que en esas edades es donde las personas están preparadas para el cambio de funciones, o de roles; como así también, hay son momentos de maduración profesional, indicado por ciertas competencias que van adquiriendo, como así también, momentos en que se producen nuevas búsquedas en función de una mejora, no solo remunerativa, sino en el reconocimientos de la tareas, cabe agregar que se observa también la migración

de grandes organizaciones a más pequeñas, y en algunos casos a emprendimientos propios, como nuevos start ups, después de haber adquirido cierta experiencia en otra organización.

2.- Resultados para las variables Edad y Localización del Equipo de desarrollo en el que se desempeña

En el apartado anterior se explicaron cuáles fueron los valores previstos para la variable Edad, en tanto que para la variable modalidad de trabajo los valores fueron: Co-locado, distribuido, mixto, principalmente co-locado . Los datos obtenidos, pusieron de manifiesto que existían algunos casos de miembros distribuidos en proyectos particulares y alternando en función de los días con ambas modalidades. Estos últimos valores, surgieron de un reagrupamiento que se efectuó al analizar los datos obtenidos.

Los resultados obtenidos pueden observarse en el Gráfico 30

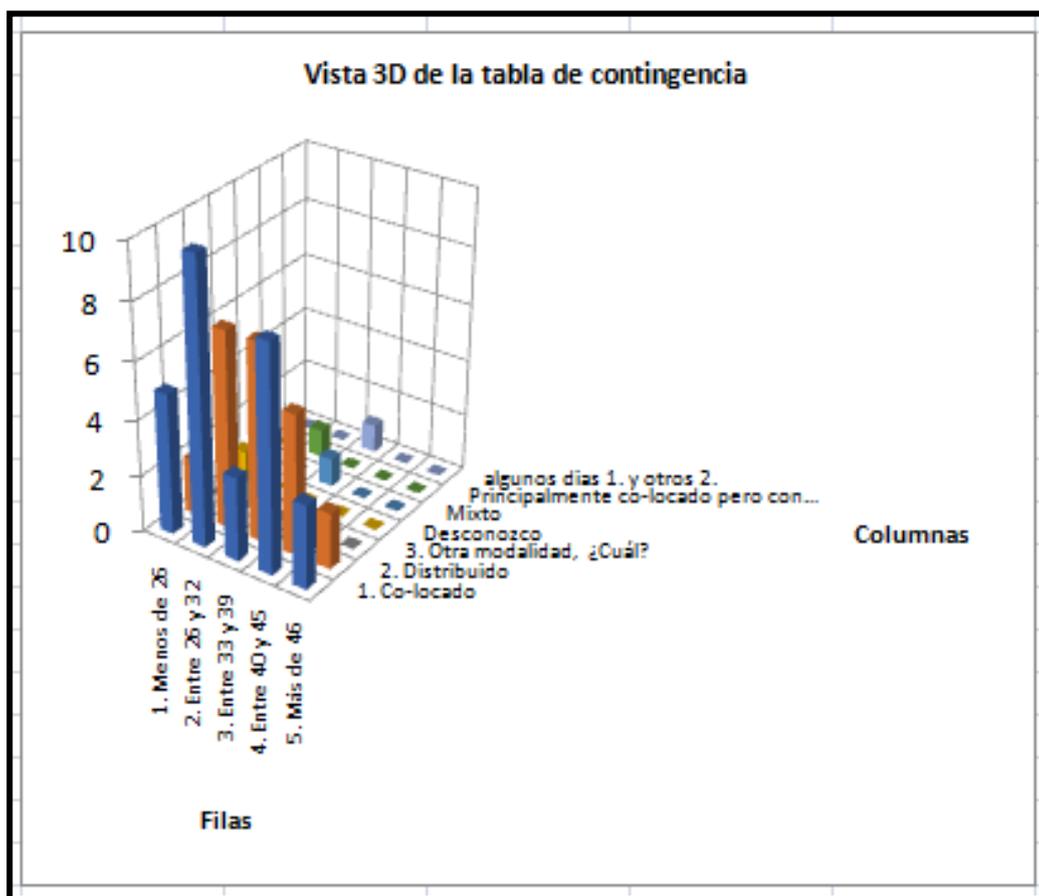


Gráfico 30. Tabla de contingencia para las variables edad y Localización del Equipo de desarrollo en el que se desempeña

A continuación se presentan los resultados de la Prueba Chi Cuadrado de Independencia. Si se observan los resultados de la Tabla 2 se advierte que existe independencia entre filas y columnas de la tabla de contingencia, en función de que el p-valor computado (0,621) es mayor que el nivel de significación alfa=0,05, por lo tanto se acepta la hipótesis nula H0 y se rechaza la Hipótesis Alternativa Ha. Cabe aclarar que el riesgo de rechazar la hipótesis nula H0 cuando es verdadera es de 62,10 %.

Se observa que la mayor ocurrencia en los rangos de edades es para equipos co-locados o sea equipos que trabajan en un mismo espacio físico, a excepción del rango etario entre los 33 y los 39 años, lo que podría deberse a que es el rango etario en el cual las personas tienen hijos pequeños por lo que prefieren la posibilidad de hacer trabajo desde la casa denominado home work.

Tabla 2. Prueba de Chi cuadrado entre edad y Localización del Equipo de desarrollo en el que se desempeña.
Fuente Elaboración Propia

Chi-cuadrado ajustado (Valor observado)	21,299
Chi-cuadrado ajustado (Valor crítico)	36,415
p-valor	0,621

3.- Resultados para las variables Antigüedad en la última función de trabajo y Localización del Equipo de desarrollo en el que se desempeña

En este caso ya se enunciaron los valores que se utilizaron para ambas variables. Los resultados obtenidos se pueden observar en el Gráfico 31.

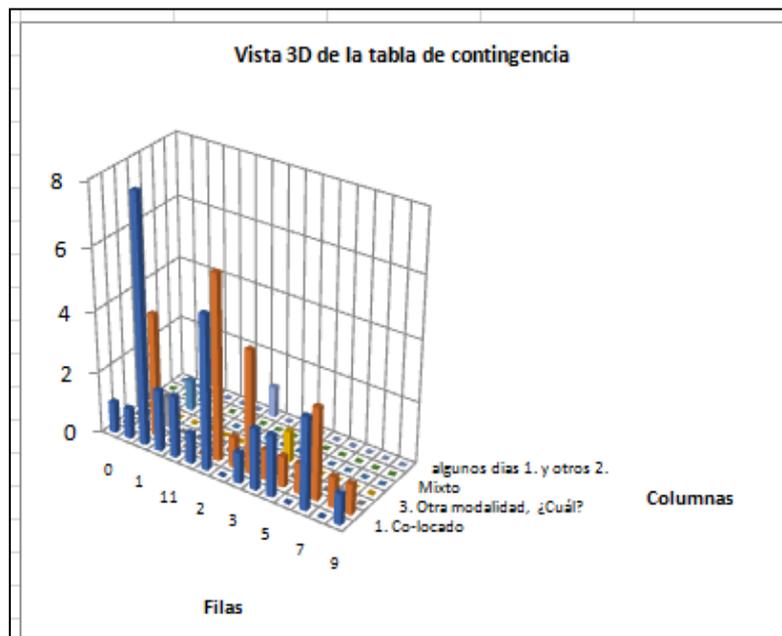


Gráfico 31. Tabla de contingencia para las variables antigüedad en la última función en el trabajo y Localización del Equipo de Desarrollo en el que se desempeña
Fuente: Elaboración Propia

Si se analizan los datos obtenidos en la prueba Chi Cuadrado de independencia, se observa que como el p-valor calculado es mayor que el nivel de significación $\alpha = 0,05$ se puede aceptar la hipótesis Nula H_0 , es decir que no existe dependencia entre filas y columnas. Cabe aclarar que el riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es de 99,28%.

Tabla 3. Prueba de Chi Cuadrado para las variables antigüedad en la última función en el trabajo y localización del Equipo de Desarrollo en el que se desempeña
Fuente: Elaboración Propia

Chi-cuadrado ajustado (Valor observado)	55,616
Chi-cuadrado ajustado (Valor crítico)	106,395
p-valor	0,993

4.- Resultados para las variables Principales tareas que realiza en el trabajo y Localización del Equipo de Desarrollo en el que se desempeña

En este caso, para la variable, Principales Tareas que realiza, se utilizaron los siguientes valores: Gestión de Proyecto, Ingeniería de requerimientos, Arquitectura, Diseño, Implementación, Testing y Despliegue. También se incluyó una respuesta abierta, para el caso de que ninguna de las opciones se ajustara a la respuesta de los encuestados. En Gráfico 32 se presenta la Tabla de Contingencia de ambas variables

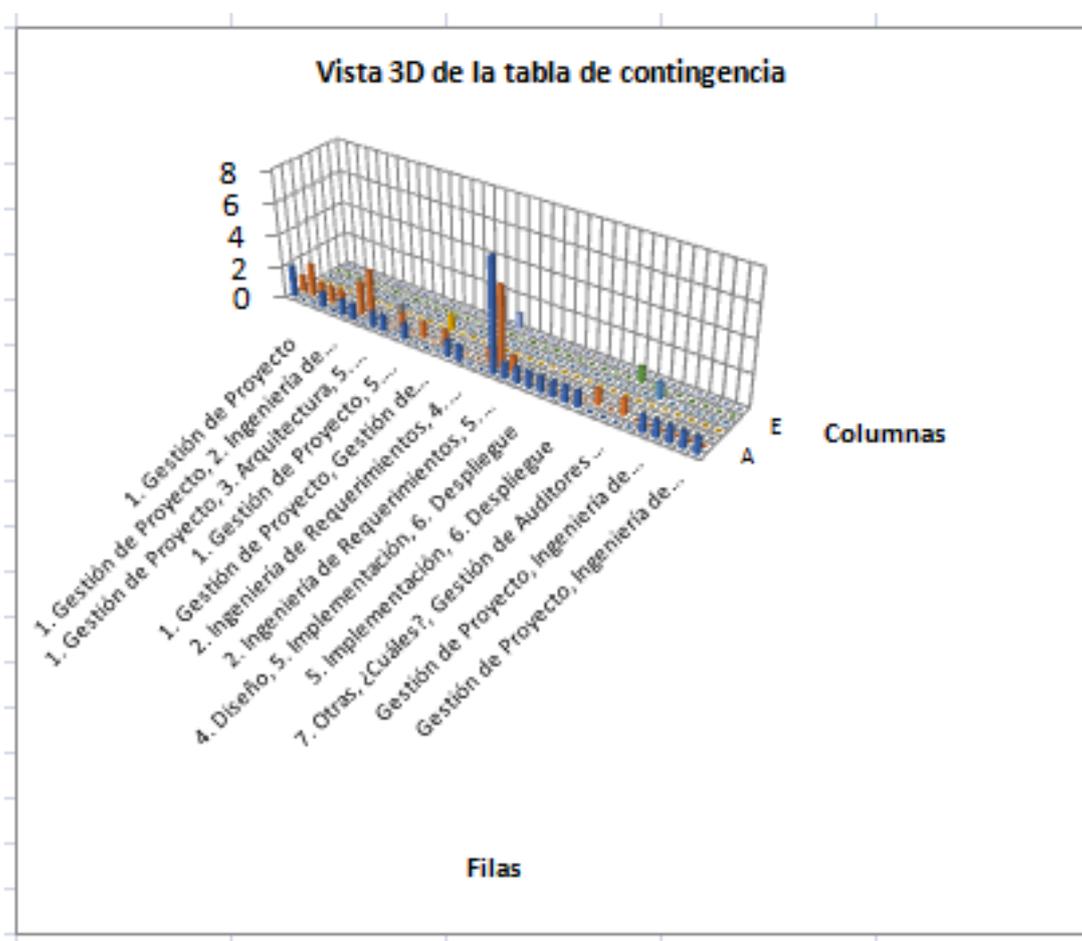


Gráfico 32. Tabla de contingencia para las variables principales tareas que realiza en el trabajo y Localización del Equipo de Desarrollo en el que se Desempeña
Fuente: Elaboración Propia

Si se observan los datos de la Tabla 4, correspondiente a la Prueba de Chi Cuadrado de Independencia, teniendo en cuenta que el p-valor calculado es menor que el nivel de significación $\alpha = 0,05$ se debe rechazar la Hipótesis Nula H_0 y acertar la Hipótesis Alternativa H_a . Es decir se puede afirmar que

existe dependencia entre las variables analizadas. En este caso, el riesgo de rechazar la Hipótesis nula H0 cuando es verdadera es menor que 0,02%.

Estos resultados serían indiciarios de que la modalidad (distribuida o co-locada) en relación con la localización de los equipos de trabajo articula con las principales tareas que cada uno desempeña. La mayor incidencia de respuestas está en el tipo de trabajo co-locado lo cual es coherente con que el desarrollo de software es un trabajo en equipo, la comunicación, el trabajo colaborativo, la cooperación son requeridos sobre todo y especialmente en las tareas de requerimientos, diseño e implementación que son las que mayor impacto tienen; las que son muy difíciles de realizar si el equipo está distribuido físicamente, es decir si no hay presencia. Para el caso específico de la gestión de proyectos es necesario destacar que es casi imprescindible la cohabitación en un mismo espacio de trabajo para poder cumplir con las actividades de gestión.

Tabla 4. Prueba de Chi cuadrado de independencia para las variables principales tareas que realiza en el trabajo y modalidad del equipo de desarrollo en el que trabaja
Fuente: Elaboración Propia

Chi-cuadrado ajustado (Valor observado)	289,187
Chi-cuadrado ajustado (Valor crítico)	244,808
p-valor	0,000

5.- Resultados para las variables Utilización de metodologías /Frameworks basados en filosofía Lean/Ágil y Localización del equipo de Desarrollo en el que se desempeña

En este apartado se presentan los resultados correspondientes a la Prueba de Independencia Chi cuadrado para las variables Utilización de metodologías basadas en filosofía Lean/Ágil y Localización del Equipo de Desarrollo en el que se desempeña. En este último caso, ya se han explicitado los valores que se utilizaron: Co-locado, distribuido, mixto y principalmente co-locado. En relación con las metodologías basadas en filosofía Lean/Ágil, los valores utilizados fueron: Scrum, XP, FDD, Kanban, Scrumban, ATDD/TDD, no utilizo metodologías Lean/Agiles. En este caso también se previó una opción de respuesta abierta.

En el Gráfico 33 se presenta la Tabla de Contingencia para las variables analizadas

Si se observan los resultados de la Prueba de Independencia Chi Cuadrado, de la Tabla 5, se puede afirmar que teniendo en cuenta que el p-valor computado es menor que el nivel de significación alfa =0,05, se debe rechazar la hipótesis nula H= y aceptar la hipótesis alternativa Ha. Es decir, existe dependencia entre ambas variables. Cabe aclarar que el riesgo de rechazar la hipótesis nula H0 cuando es verdadera es menor que 0,04%. En este caso el resultado podría estar mostrando que: dado que la mayoría de los encuestados trabaja con SCRUM como framework de gestión de proyectos, la opción de localización elegida es la presencial. Esto se fundamenta en la recomendación que hace el framework, respecto a los ambientes de trabajo, frecuentemente denominados *scrum zone*, que plantea todo el equipo trabajando junto en el

mismo espacio, colaborando y cooperando entre sí y con el medio de comunicación cara a cara por excelencia.

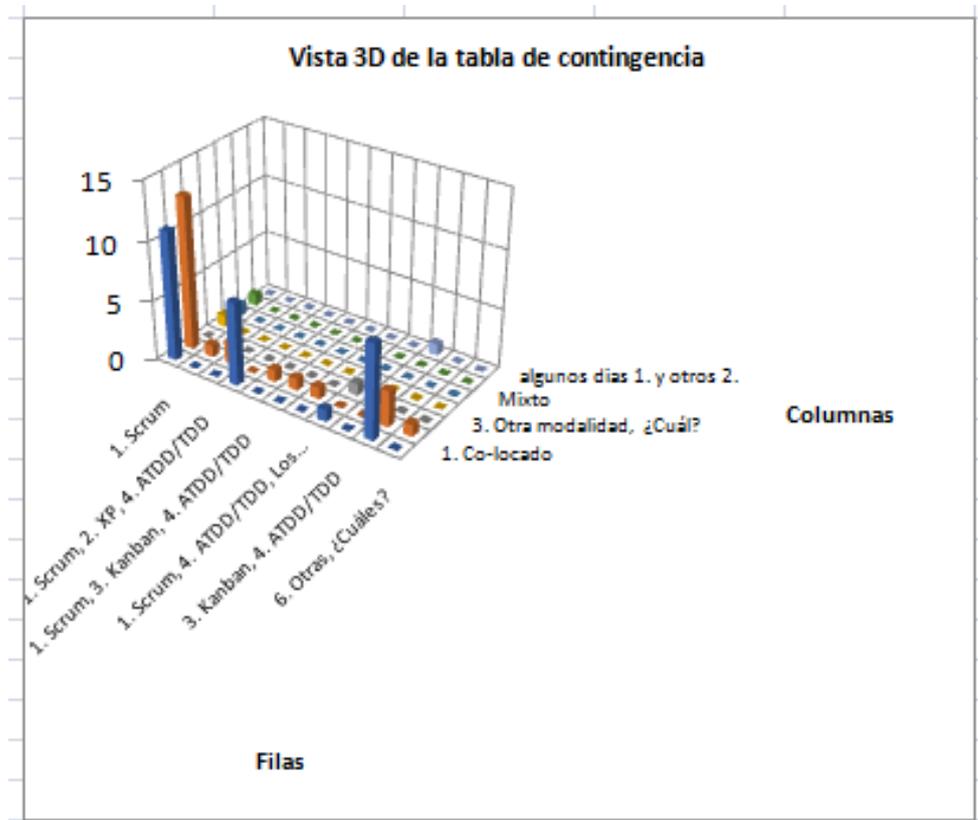


Gráfico 33. Tabla de contingencia para las variables metodologías/frameworks basadas en filosofía Ágil y Localización del equipo de Desarrollo en el que se desempeña
Fuente: Elaboración Propia

Tabla 5. Prueba de Chi Cuadrado correspondiente a las variables metodologías/frameworks basadas en filosofía Ágil y Localización del equipo de Desarrollo en el que se desempeña
Fuente: Elaboración Propia

Chi-cuadrado ajustado (Valor observado)	103,533
Chi-cuadrado ajustado (Valor crítico)	79,082
p-valor	0,000

6.- Resultados para las variables Metodología de gestión de proyectos que utiliza el equipo de desarrollo en el que trabaja y Localización del Equipo de Desarrollo en el que se desempeña

En este apartado se analizan los resultados para la variable metodología de Gestión de proyectos que utiliza el equipo de desarrollo en el que los encuestados trabajaban. Los valores para dicha variable fueron: Metodología basada en PMI, en CMMI, Ágil, Lean, basada en RUP, o no se utilizan metodologías de gestión

de Proyectos. Por último se incluyó una opción abierta en el caso de que no se utilizara ninguna de las alternativas de gestión previstas.

Para la variable Localización del equipo de Desarrollo en el que se desempeña, en los apartados anteriores, ya se explicitaron los valores con los que se trabajó (Co-Locado y Distribuido fundamentalmente).

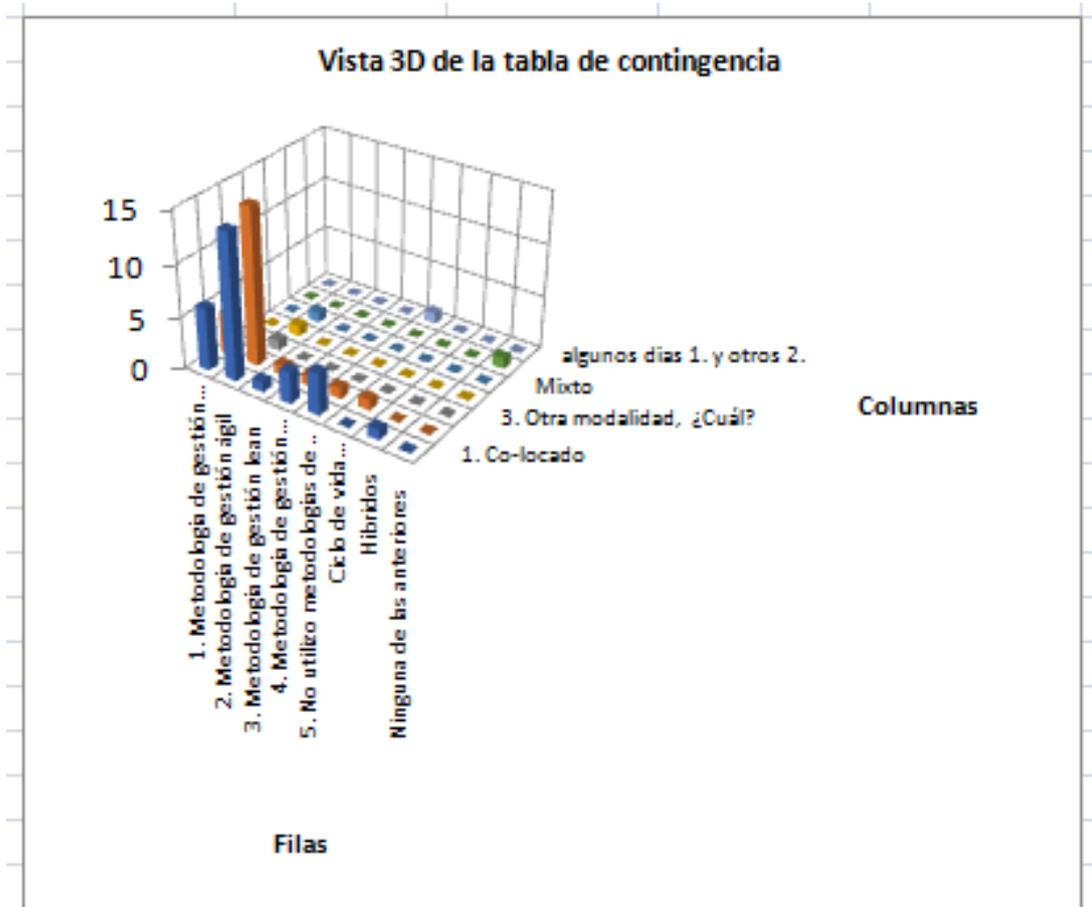


Gráfico 34. Tabla de Contingencia para las variables, metodología de gestión de proyectos que utiliza el equipo de desarrollo y Localización del Equipo de desarrollo en el que se desempeña
Fuente: Elaboración propia

En el caso de las variables que se analizan, si se observan los resultados de la prueba de Chi Cuadrado de la una confirmación del análisis realizado en el gráfico anterior, puesto que las metodologías de gestión de proyectos ágiles recomiendan para los equipos de desarrollo la localización, trabajo del equipo en un mismo medio ambiente.

Tabla 6 y teniendo en cuenta que el p-valor computado es menor que el nivel de significación alfa =0,05, se debe rechazar la hipótesis nula H_0 y aceptar la hipótesis alternativa H_a . Es decir, se puede afirmar que existe dependencia entre ambas variables. Cabe aclarar que el riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es menor que 0,23%. Este resultado sería una confirmación del análisis realizado en el gráfico anterior, puesto que las metodologías de gestión de proyectos ágiles recomiendan para los equipos de desarrollo la localización, trabajo del equipo en un mismo medio ambiente.

Tabla 6. Prueba de Chi Cuadrado de independencia para las variables metodología de gestión de proyectos que utiliza el equipo de desarrollo y Localización del Equipo de desarrollo en el que se desempeña
Fuente: Elaboración Propia

Chi-cuadrado ajustado (Valor observado)	72,711
Chi-cuadrado ajustado (Valor crítico)	58,124
p-valor	0,002

7.- Resultados para las variables Competencias Técnicas requeridas en los equipos de Localización del Equipo de Desarrollo en el que se desempeña

En el caso de la variable que se analiza, para la variable Competencias Técnicas se utilizaron los siguientes valores: Programación o Desarrollo, Técnicas de Ingeniería de requerimientos (Casos de Uso, Técnicas de Elicitación), Gestión Ágil de Requerimientos (user stories), Testing de Software, Gestión de Personas, Gestión Ágil de Proyectos, Gestión Tradicional de Proyectos, Modelado de Arquitectura de Software, Diseño de Interacción Humano-Máquina, Manejo de Herramientas para Testing Automatizado y Manejo de herramientas para Gestión de Configuración/Integración Continua. Por último y para el caso de que el encuestado considera otra competencia técnica requerida, se dejó una opción de respuesta abierta. En el Gráfico 35 se presenta la Tabla de Contingencia para ambas variables.

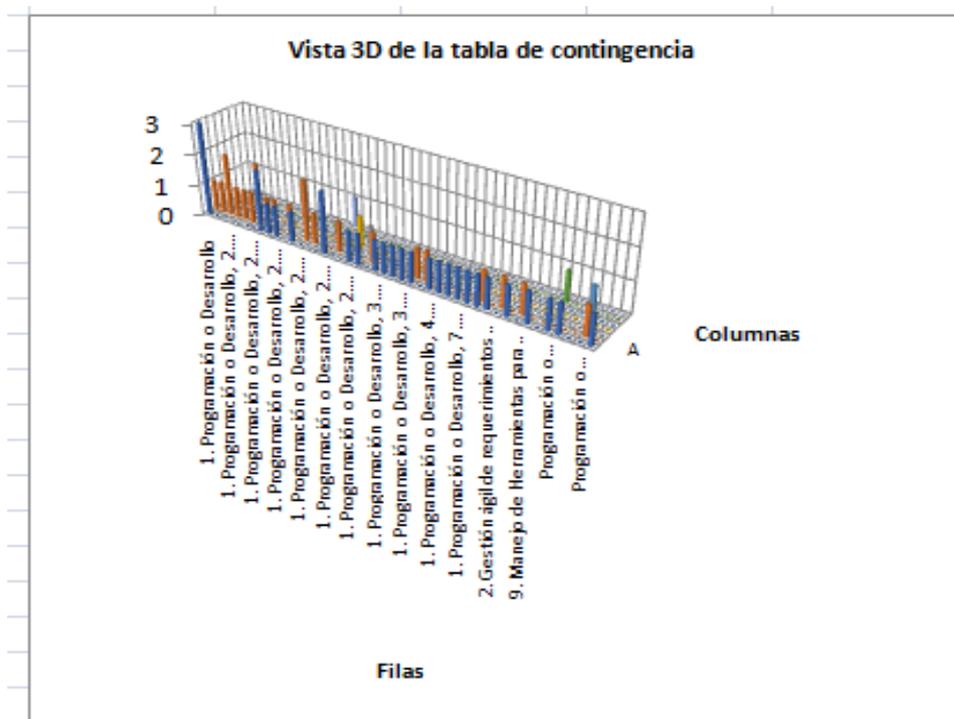


Gráfico 35. Tabla de Contingencia para las variables Competencias Técnicas requeridas en los equipos de Localización del Equipo de Desarrollo en el que se desempeña
Fuente: Elaboración propia

Si se observan los datos de la Tabla 7 y teniendo en cuenta que el p-valor computado es menor que el nivel de significación $\alpha=0,05$, se debe rechazar la hipótesis nula H_0 , y aceptar la hipótesis alternativa H_a . Por lo tanto se concluye que existe entre las variables relación de dependencia. Cabe aclarar que riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es menor que 3,85%. Este resultado es indicativo que la mayoría de estas Competencias Técnicas que apuntan a la programación o desarrollo en primer medida y al resto de las actividades de ingeniería después, realizada en el contexto de framework de gestión ágil, con llevan a la selección de equipos co-locados.

Tabla 7. Prueba de Chi Cuadrado para las variables Competencias Técnicas requeridas en los equipos de Localización del Equipo de Desarrollo en el que se desempeña
Fuente: Elaboración Propia

Chi-cuadrado ajustado (Valor observado)	299,556
Chi-cuadrado ajustado (Valor crítico)	296,466
p-valor	0,038

8.- Resultados para las variables tareas que realiza y sexo

En este caso se trabajó con la variable principales tareas que cada encuestado realiza y la variable sexo. A continuación se presenta la tabla de contingencia (ver Gráfico 36)

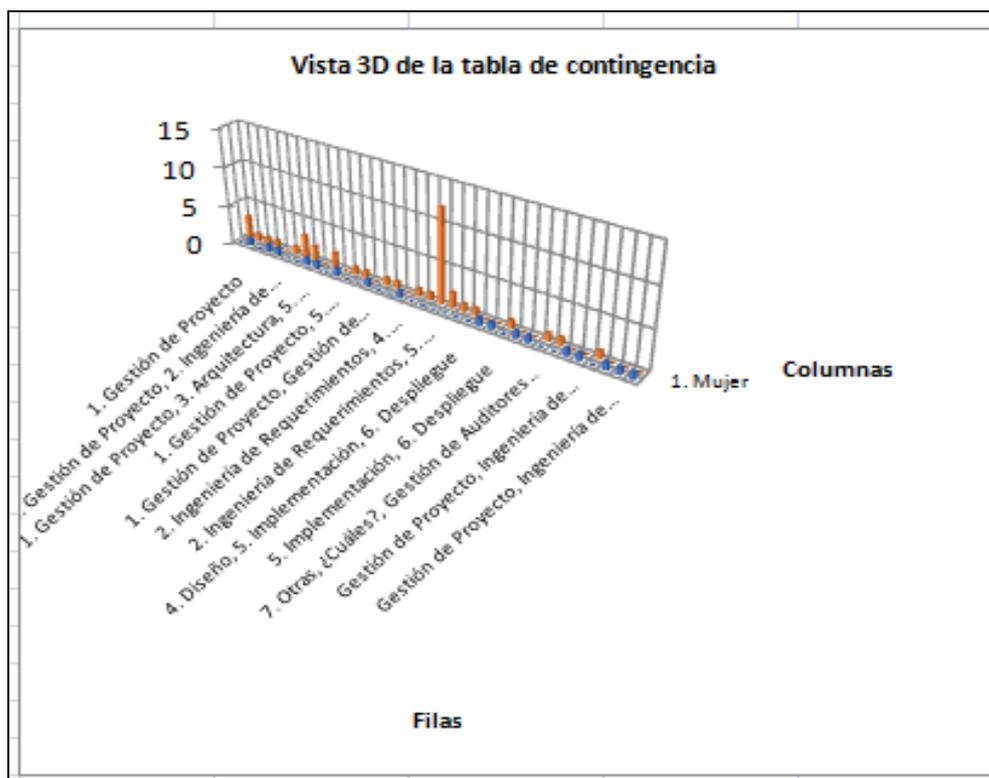


Gráfico 36. Tabla de Contingencia para las variables Tareas que Desarrolla y Sexo
Fuente: Elaboración propia

A continuación en la Tabla 8 se presentan los resultados de la prueba Chi Cuadrado de Independencia.

Tabla 8. Prueba de Chi Cuadrado de independencia para las variables principales tareas que realiza y sexo
Fuente: Elaboración propia

Chi-cuadrado ajustado (Valor observado)	49,037
Chi-cuadrado ajustado (Valor crítico)	49,802
p-valor	0,058

En este caso se observa que el p-valor calculado es mayor que el nivel de significación $\alpha = 0,05$ por lo tanto se puede aceptar la hipótesis nula H_0 , por lo que se puede afirmar que ambas variables son independientes. Cabe aclarar que el riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es de 5,80 %. Este resultado permite afirmar que no existe dependencia entre ambas variables, por lo que las tareas que se realizan en los ámbitos laborales de la Informática no están relacionados al sexo de quien las realizan

9.- Resultados para las variables principales tareas que desarrolla en el trabajo y utilización de metodologías basadas en procesos definidos

En este apartado se presentan los resultados de la prueba de independencia Chi Cuadrado para las variables principales tareas que desarrolla en su trabajo por un lado y si utiliza metodologías basadas en proceso definidos. En el primer caso, los valores con los que se trabajó fueron: Gestión de Proyecto, ingeniería de Requerimientos, Arquitectura, Diseño, Implementación, Testing y Despliegue. También existió una alternativa de respuesta abierta. En relación con la variable, uso de metodologías basadas en procesos definidos, los valores para la misma fueron: Proceso Unificado de Desarrollo (PUD), Rational Unified Process (RUP), ICONIX, Yourdon, Microsoft Solution Framework, no se utilizan metodologías basadas en procesos definidos y otras metodologías (opción de respuesta abierta).

En el Gráfico 37 se presenta la tabla de contingencia para dichas variables.

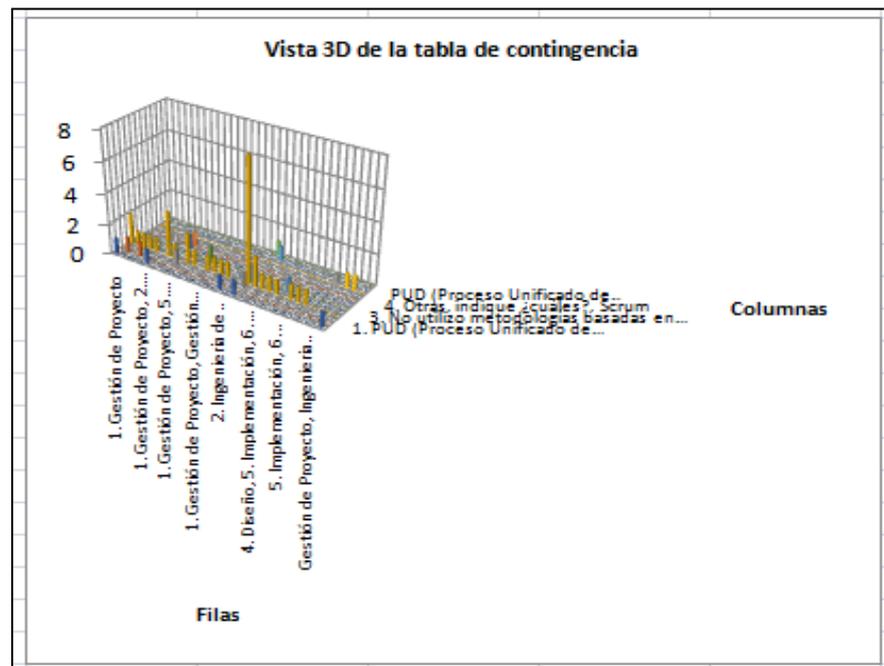


Gráfico 37. Tabla de Contingencia para las variables Tareas que Desarrolla y utilización de metodologías basadas en procesos definidos
Fuente: Elaboración propia

A partir de los datos que se presentan en la Tabla 9 se puede observar que para estas variables el p-valor calculado es mayor que el nivel de significación $\alpha=0,05$ por lo que se puede aceptar la hipótesis nula H_0 y rechazar la Hipótesis Alternativa H_a . En este caso el riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es de 14,71%. En este caso, y a partir de los datos obtenidos en el presente estudio, se puede afirmar que ambas variables, tareas que se realizan y metodologías basadas en procesos definidos, tienen un funcionamiento independiente una de la otra.

Tabla 9. Prueba Chi Cuadrado para las variables Tareas que Desarrolla y utilización de metodologías basadas en procesos definidos
Fuente: Elaboración propia

Chi-cuadrado ajustado (Valor observado)	356,988
Chi-cuadrado ajustado (Valor crítico)	373,363
p-valor	0,147

10.- Resultados para las variables principales tareas que desarrolla y competencias conductuales requeridas

En este apartado, se analizan las variables principales tareas que desarrolla en el trabajo, cuyos valores ya se explicitaron y competencias conductuales requeridas, para la cual se adoptaron los siguientes valores: resolución de problemas complejos, Trabajo en equipo, Autoorganización, manejo de Inglés, pensamiento Crítico, Inteligencia Emocional, Orientación a Servicios, negociación y Flexibilidad Cognitiva, con la posibilidad de una última opción abierta, cuyos resultados se presentan en el Gráfico 38.

11.- Resultados para las variables principales tareas que se desarrollan y aspectos de la relación vincular entre los involucrados en el desarrollo de un producto de software que son necesarios mejorar.

En este apartado se presentan los resultados del análisis bivariado, correspondiente a las variables, Tareas que Desarrolla y aspectos de la relación vincular que en el desarrollo de un producto de software es necesario mejorar. Para esta última variable, los valores con los que se trabajó son: comunicación, compromiso, Involucramiento, Cooperación y una opción de respuesta abierta.

En el Gráfico 39 se presenta la Tabla de Contingencia correspondiente a dichas variables.

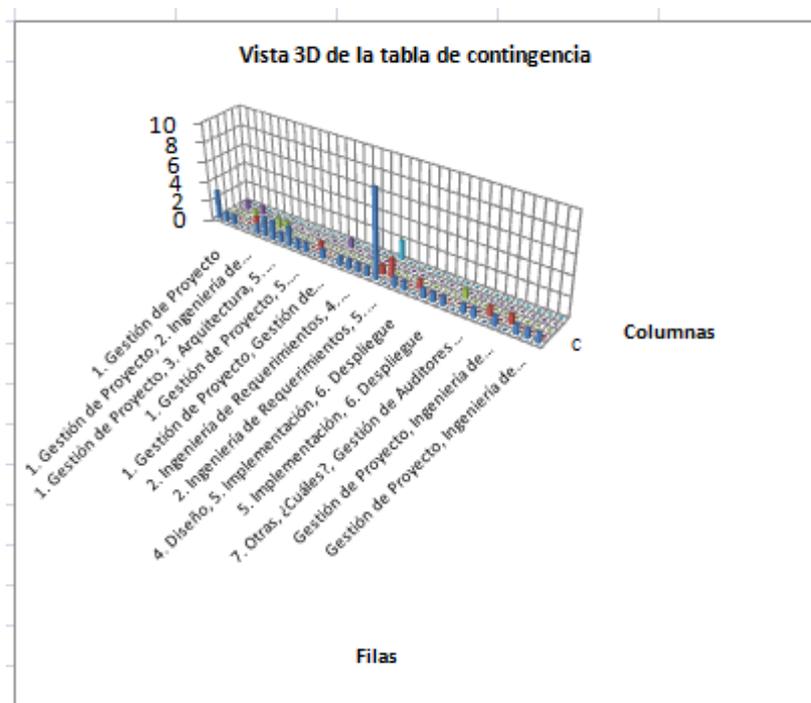


Gráfico 39. Tabla de contingencia para las variables, Tareas que desarrolla y aspectos de la relación vincular que es necesario mejorar
Fuente: Elaboración Propia

Si se observan los datos que se presentan en la Tabla 11, correspondientes a la Prueba Chi Cuadrado de Independencia para las variables, se advierte que el P-valor calculado (0,846) resulta mayor que el valor de significancia $\alpha = 0,05$. Por tal motivo se puede concluir que no existe relación de dependencia entre las variables que se analizan. En este caso en los que los resultados, indican que se puede aceptar la hipótesis nula H_0 y rechazar la Hipótesis Alternativa H_a , el riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es de 84,5%.

El resultado de independencia entre ambas variables permite afirmar que no existe relación alguna entre el tipo de trabajo que se desarrolla y los aspectos relacionales a mejorar, por lo tanto podría considerarse que ellos son importantes y no vinculantes a una tarea específica; es decir en todas las

funciones requeridas para crear software son demandados los aspectos de relación vincular, en especial la comunicación y la cooperación, en un segundo orden el compromiso y el involucramiento.

Tabla 11. Prueba de chi cuadrado para las variables Tareas que desarrolla y aspectos de la relación vincular que es necesario mejorar
Fuente: Elaboración propia

Chi-cuadrado ajustado (Valor observado)	123,025
Chi-cuadrado ajustado (Valor crítico)	168,613
p-valor	0,846

12.- Resultados para las variables competencias conductuales requeridas en los equipos de trabajo y competencias técnicas son requeridas

En este apartado se indaga la existencia de relación entre competencias conductuales y técnicas requeridas en los distintos equipos de trabajo. Para las competencias conductuales, se trabajó con los siguientes valores: resolución de problemas complejos, Trabajo en Equipo, Autoorganización, manejo de Ingles, Pensamiento Crítico, Inteligencia Emocional, Orientación a Servicios, Negociación y Flexibilidad Cognitiva, con una última opción de respuesta abierta. En el caso de las competencias técnicas los valores de la variable fueron: Programación o desarrollo, manejo de Técnicas de Ingeniería de requerimientos, Gestión Ágil de requerimientos, Testing de Software, Gestión de Personas, Gestión Ágil de proyectos, Gestión Tradicional de Proyectos, modelado de Arquitectura de Software, Diseño de Interacción Humano Maquina, manejo de Herramientas para Testing Automatizado y manejo de Herramientas para gestión de Configuración /integración Continua. Por último en este caso también existió una alternativa de respuesta abierta. En el Gráfico 40 se presenta la Tabla de Contingencia correspondiente al análisis bivariado.

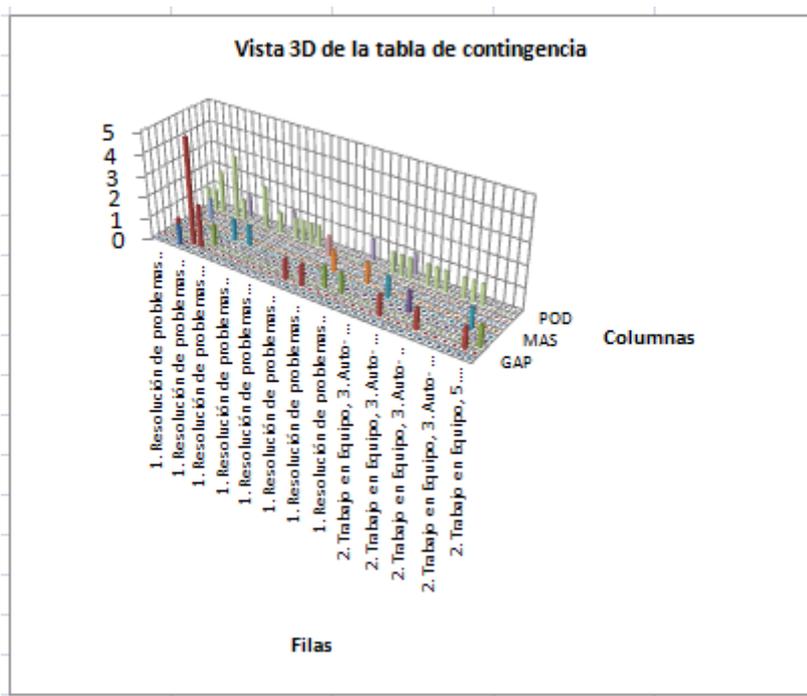


Gráfico 40. Tabla de contingencia para las variables competencias conductuales requeridas en los equipos de trabajo y competencias técnicas son requeridas
Fuente: Elaboración propia

En la Tabla 12 se presentan los datos correspondientes a la Prueba de Independencia chi Cuadrado. De la lectura de estos, se observa que el p-valor calculado (0,285), es mayor que el nivel de significación alfa = 0,05, por lo que se puede aceptar la hipótesis nula H_0 y rechazar la hipótesis alternativa H_a . Esto implica que de acuerdo con los datos obtenidos, no existe relación entre ambas variables. Cabe aclarar que en este caso, el riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es de 28,51%. Estos resultados, resultan indiciarios que al no existir relación entre ambos tipos de competencias, los aspectos técnicos y conductuales no tienen relación entre ellos. No obstante dado que el desarrollo de software es una actividad que se realiza principalmente en equipo, o sea el grupo está centrado en la tarea de crear y desarrollar software; cada una las competencias técnicas involucradas en esa creación son importantes y diferentes, para la ejecución de todas ellas es necesaria la presencia de **todas** las competencias conductuales analizadas destacando muy especialmente las relacionadas con trabajo en equipo autoorganización y resolución de problemas complejos, las que deben estar presentes en forma constante y permanente para que el equipo de trabajo alcance las metas propuestas.

Tabla 12. Prueba de chi cuadrado para las variables competencias conductuales requeridas en los equipos de trabajo y competencias técnicas son requeridas
Fuente: Elaboración propia

Chi-cuadrado ajustado (Valor observado)	347,184
Chi-cuadrado ajustado (Valor crítico)	376,555
p-valor	0,285

13.- Resultados para las variables competencia conductuales requeridas y aspectos de la relación vincular que es necesario mejorar

En el presente apartado se analiza la relación entre competencias conductuales requeridas y aspectos de la relación vincular que es necesario mejorar. En ambos casos, ya se han explicitado los valores con los que se trabajó. En el Gráfico 41 se presenta la Tabla de Contingencia correspondiente a dichas variables.

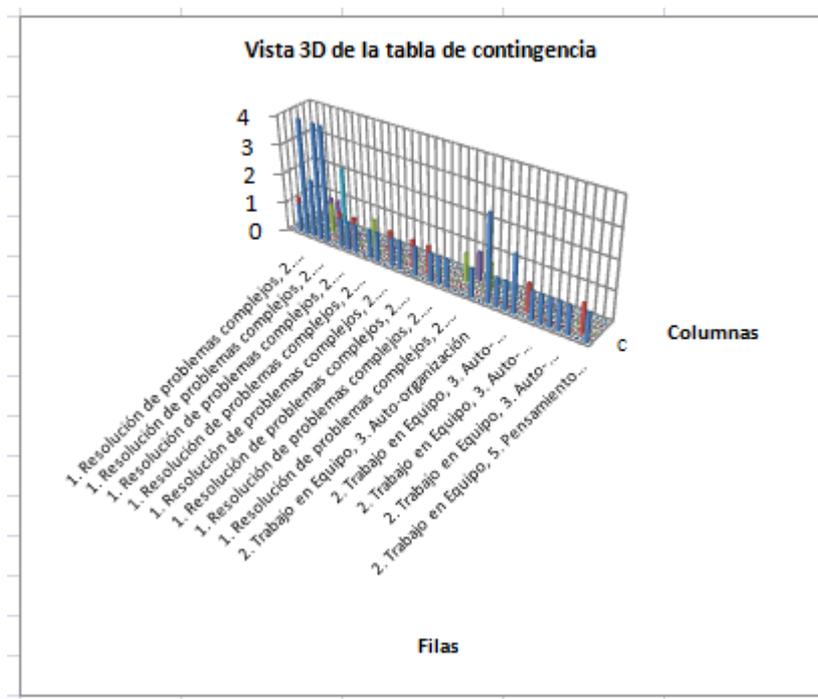


Gráfico 41. Tabla de Contingencia para las variables, competencias conductuales requeridas y aspectos de la relación vincular a mejorar.
Fuente: Elaboración propia

En la Tabla 13 correspondiente a la prueba de Independencia Chi Cuadrado, se observa que el P-valor calculado (0,762) es mayor que el nivel significancia alfa 0,05, por lo que resulta aconsejable aceptar la hipótesis nula H_0 y rechazar la hipótesis alternativa H_a , teniendo en cuenta que en este caso el riesgo de rechazar la hipótesis nula H_0 cuando es verdadera es de 76,25%. Por tal motivo, y de acuerdo a los resultados obtenidos, se infiere que no existe relación directa entre cada una de las competencias conductuales requeridas y los aspectos relacionales a mejorar. Sin embargo como en el caso anterior, se considera que los aspectos a mejorar más frecuentes son en primer lugar la comunicación en forma muy destacada, luego la cooperación y en tercer lugar compromiso e involucramiento, al contrastarlo contra las competencias conductuales se observa que las más requeridas son el trabajo en equipo, la autoorganización y la resolución de problemas complejos; ninguna de estas es viable sin la presencia de los aspectos destacados recientemente como a mejorar.

Tabla 13. Prueba Chi cuadrado de Independencia para las variables, competencias conductuales requeridas y aspectos de la relación vincular a mejorar.
Fuente: Elaboración propia

Chi-cuadrado ajustado (Valor observado)	135,415
Chi-cuadrado ajustado (Valor crítico)	177,390
p-valor	0,762

En síntesis, y a partir de los resultados obtenidos con la prueba de Independencia de Chi Cuadrado, se observan las siguientes relaciones de dependencia: (Ver Tabla 14)

Tabla 14. Relaciones de Dependencia surgidas de la Prueba de Independencia Chi Cuadrado
Fuente: Elaboración Propia

Variable 1	Variable 2	P- valor	Relación
Edad	Antigüedad	0,024	Dependencia
Edad	Localización del equipo de desarrollo en el que se desempeña	0,621	Independencia
Antigüedad	Localización del equipo de desarrollo en el que se desempeña	0,993	Independencia
Principales tareas que realiza	Localización del equipo de desarrollo en el que se desempeña	0,000	Dependencia
Utilización de metodologías/Framework basadas en filosofía Lean/Agiles	Localización del equipo de desarrollo en el que se desempeña	0,000	Dependencia
Metodología de Gestión de proyectos que se utiliza	Localización del equipo de desarrollo en el que se desempeña	0,002	Dependencia
Competencias técnicas Requeridas	Localización del equipo de desarrollo en el que se desempeña	0,038	Dependencia
Tareas que realiza	Sexo	0,058	Independencia
Tareas que realiza	Metodologías Procesos Definidos	0,147	Independencia
Tareas que realiza	Competencias Conductuales	0,003	Dependencia
Tareas que realiza	Aspectos de la relación Vincular a mejorar	0,846	Independencia
Competencias Conductuales	Competencias Técnicas	0,285	Independencia
Competencias Conductuales	Aspectos de la relación vincular a mejorar	0,762	Independencia

Análisis Multivariado

Al realizar el análisis multivariado, de los resultados obtenidos en el caso de la variable competencias técnicas requeridas, se obtuvieron los datos de la Tabla 15 para las estadísticas descriptivas. Los resultados obtenidos muestran que la mayor media alcanzada corresponde a programación o Desarrollo (0,914), seguido por Testing de Software (0,655) y en tercer lugar Gestión Ágil de requerimientos con una media de 0,586.

Tabla 15. Estadísticas descriptivas para la variable Competencias Técnicas requeridas
Fuente: Elaboración propia

Variable	Observaciones	Media	Desviación típica
Programación o desarrollo POD	58	0,914	0,283
Manejo de Técnicas de Ingeniería de requerimientos MTI	58	0,034	0,184
Gestión Ágil de requerimientos GAR	58	0,586	0,497
Testing de Software TS	58	0,655	0,479
Gestión Ágil de Proyectos GAP	58	0,466	0,503
Gestión tradicional de Proyectos GTP	58	0,207	0,409
Gestión de Personas GP	58	0,431	0,500
Modelado de Arquitectura de Software MAS	58	0,483	0,504
Diseño de Interacción Humano Maquina DI	58	0,207	0,409
Manejo de Herramientas para el Testing Automatizado MHTA	58	0,052	0,223
Manejo de Herramientas para la Gestión de Configuración Integración Continua MH	58	0,483	0,504

Por otra parte, si se analizan los datos correspondientes a la matriz de Correlación Pearson (Tabla 16) se observan las siguientes correlaciones:

Tabla 16. Matriz de correlación Pearson correspondiente a la variable Competencias Técnicas
Fuente: Elaboración Propia

Matriz de correlación (Pearson (n)):

Variables	POD	MTI	GAR	TS	GAP	GTP	GP	MAS	DI	MHTA	MH
POD	1	-0,279	0,116	0,294	0,163	-0,450	0,019	0,174	0,005	0,072	0,174
MTI	-0,279	1	0,159	0,137	0,013	0,370	0,217	0,007	0,137	0,383	0,007
GAR	0,116	0,159	1	0,422	0,433	-0,003	0,307	-0,169	0,256	0,196	0,181
TS	0,294	0,137	0,422	1	0,313	0,012	0,265	0,193	0,102	0,169	0,048
GAP	0,163	0,013	0,433	0,313	1	0,121	0,235	0,274	0,291	0,094	0,205
GTP	-0,450	0,370	-0,003	0,012	0,121	1	0,329	0,018	0,159	0,073	0,018
GP	0,019	0,217	0,307	0,265	0,235	0,329	1	0,274	0,157	0,111	0,204
MAS	0,174	0,007	-0,169	0,193	0,274	0,018	0,274	1	0,188	0,086	0,240
DI	0,005	0,137	0,256	0,102	0,291	0,159	0,157	0,188	1	0,265	0,188
MHTA	0,072	0,383	0,196	0,169	0,094	0,073	0,111	0,086	0,265	1	0,086
MH	0,174	0,007	0,181	0,048	0,205	0,018	0,204	0,240	0,188	0,086	1

Tabla 17. Correlaciones positivas y negativas para la variable Requerimientos Técnicos
Fuente: Elaboración propia

Variable 1	Variable 2	Coefficiente de correlación r
Gestión Tradicional de Proyectos	Programación o desarrollo	- 0,450
Gestión Tradicional de Proyectos	Manejo de Técnicas de Ing. de requerimientos	0,370
Manejo de Herramientas para Testing Automatizado	Manejo de Técnicas de Ing. de requerimientos	0,383
Gestión Ágil de requerimientos	Gestión Ágil de proyectos	0,433
Gestión Ágil de Requerimientos	Testing de Software	0,422
Gestión Ágil de Requerimientos	Gestión de Personas	0,307

Los datos de la Tabla 17 muestran seis correlaciones de mediana intensidad (entre 0,300 y 450) todas ellas positivas con excepción de la correspondiente a Gestión Tradicional de Proyectos con programación o Desarrollo que presenta un valor negativo

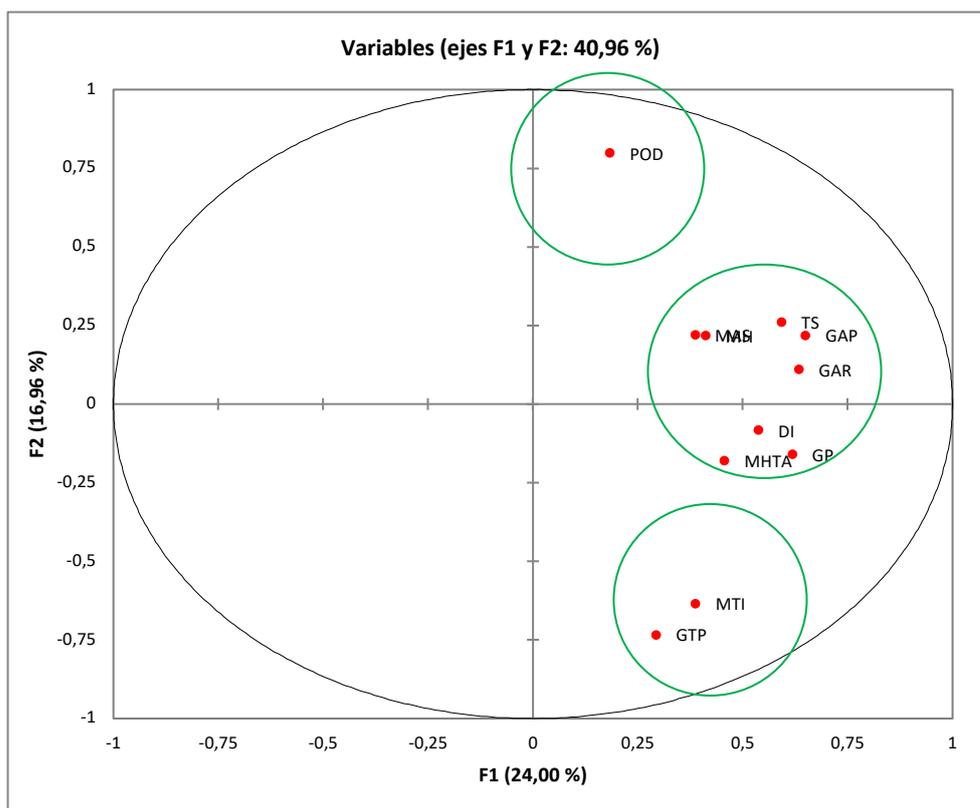


Gráfico 42. Tipologías a partir de competencias técnicas requeridas
Fuente: Elaboración propia

El Gráfico 42 presenta los resultados del análisis multivariado realizado para la variable Competencias Técnicas requeridas. Al respecto se observan dos agrupamientos bien diferenciados: por un lado las correspondientes a Programación o Desarrollo (POD) y por otro las correspondientes a Manejo de Técnicas de Ingeniería de Requerimientos, tales como casos de Uso, Técnicas de Elicitación, etc.) Junto a las de Gestión Tradicional de Proyecto. El resto de las competencias técnicas se agrupan en el plano en forma no diferenciada unas de otras. Estos resultados permiten afirmar que dentro de las competencias técnicas la que más se requiere es la vinculada a la programación lo cual es evidente, dado que de todas las actividades afectadas a la construcción de software es la única no prescindible.

La no presencia explícita de las demás trae como consecuencia la obtención de productos de software de baja calidad que no satisfacen las expectativas y necesidades de quienes los solicitaron. Desde que se comenzó a desarrollar software, la actividad que se ejecutaba por excelencia era la programación, el resto de las actividades técnicas como ser requerimientos, análisis y diseño, testing, etc. Estaban embebidas en el trabajo de la programación no quedando evidencia o materialización de su ejecución con el tiempo; el surgimiento de metodologías y procesos para el desarrollo de software propuso la discriminación, formalizando y jerarquización de estas actividades, no obstante el surgimiento de estos métodos de trabajo, no siempre fueron incorporados en las organizaciones o en sus equipos de trabajo estos no dedicaron el tiempo necesario a las actividades de ingeniería previas a la programación y a las actividades de verificación y validación posteriores a la programación.

Con respecto a la gestión del proyecto de desarrollo que obtiene como resultado el producto software en la mayoría de las organizaciones es un trabajo subestimado y considerado prescindible, lo que trae como consecuencia proyectos con presupuestos, costos y tiempos de entrega desfasados con los compromisos previstos.

Análisis Factorial

Con el objeto de refinar nuestro análisis se procedió a trabajar con técnicas estadísticas de análisis Factorial.

Variable Tareas que se realizan

En la Tabla 18 se observan las frecuencias y medias obtenidas para cada una de las variables asociadas a la tarea que los encuestados desarrollaban.

Tabla 18. Estadísticas Descriptivas variable tareas que desarrolla

Fuente. Elaboración propia

Variable	Categorías	Frecuencias	%
Gestión de proyecto	No	27	47,368
	Si	30	52,632
Ingeniería de requerimientos	No	39	68,421
	Si	18	31,579
Ingeniería de procesos	No	54	94,737
	Si	3	5,263
Arquitectura	No	35	61,404
	Si	22	38,596
Despliegue	No	45	78,947
	Si	12	21,053
Diseño	No	32	56,140
	Si	25	43,860
Implementación	No	23	40,351
	Si	34	59,649
Gestión de auditores externos	No	56	98,246
	Si	1	1,754
Testing	No	55	96,491
	Si	2	3,509
Quality assurance	No	56	98,246
	Si	1	1,754
Consultoría	No	56	98,246
	Si	1	1,754

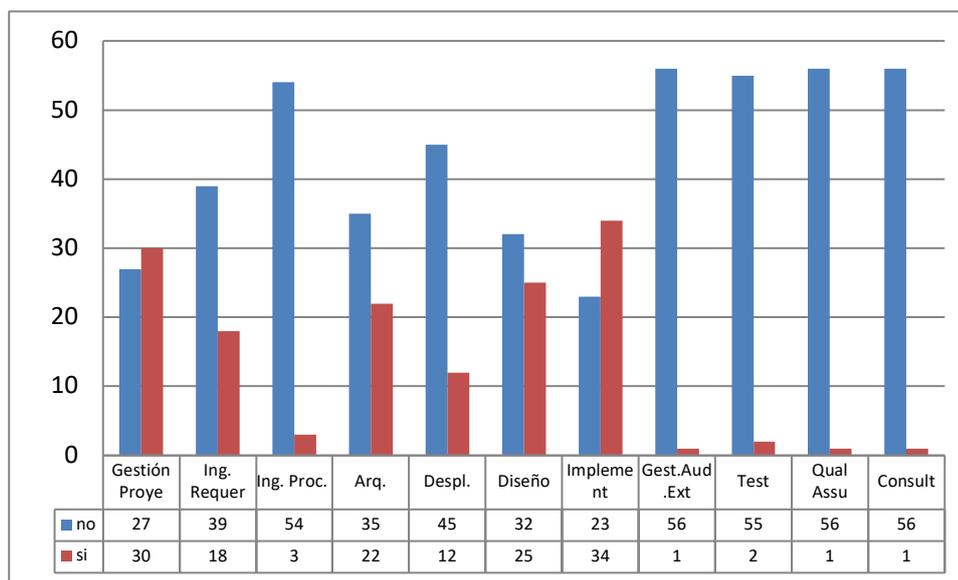


Gráfico 43. Tareas que se desarrollan

Fuente: Elaboración propia

El Gráfico 43 y la Tabla 18, muestran que las tareas con mayor desarrollo entre los encuestados corresponden Implementación (59,64%), seguida por Gestión de proyectos (52,63 %), luego por Diseño y Arquitectura (43,86 % y 38,59 % respectivamente) y por último Ingeniería de requerimientos (31,57%). Por otra

parte las tareas menos realizadas corresponden a Consultoría, Quality Assurance y Gestión Auditoría Externa (con 1,75% en cada uno de los casos)

A continuación se presenta la Tabla 19, a partir de la cual se realizó el análisis factorial, para la variable tareas que se desarrollan.

Tabla 19. Análisis Factorial. Tabla de Burt

		Gest. Proy.		Ing. Requerim		Ing. Proc		Arq		Desplieg		Diseño		Implem		Gest. audit ext-		Test		Qual. Assur		Consul	
		No	Si	No	Si	No	Si	No	Si	No	Si	No	Si	No	Si	No	Si	No	Si	No	Si	No	si
Gest. Proy.	No	27	0	21	6	26	1	14	13	20	7	9	18	5	22	27	0	26	1	26	1	27	0
	Si	0	30	18	12	28	2	21	9	25	5	23	7	18	12	29	1	29	1	30	0	29	1
Ing. Requerim	No	21	18	39	0	38	1	20	19	30	9	20	19	10	29	38	1	38	1	39	0	38	1
	Si	6	12	0	18	16	2	15	3	15	3	12	6	13	5	18	0	17	1	17	1	18	0
Ing proc	No	26	28	38	16	54	0	32	22	43	11	29	25	20	34	53	1	52	2	53	1	53	1
	Si	1	2	1	2	0	3	3	0	2	1	3	0	3	0	3	0	3	0	3	0	3	0
Arq	No	14	21	20	15	32	3	35	0	25	10	23	12	17	18	35	0	33	2	34	1	34	1
	Si	13	9	19	3	22	0	0	22	20	2	9	13	6	16	21	1	22	0	22	0	22	0
Desplieg	No	20	25	30	15	43	2	25	20	45	0	22	23	18	27	44	1	43	2	44	1	44	1
	Si	7	5	9	3	11	1	10	2	0	12	10	2	5	7	12	0	12	0	12	0	12	0
Diseño	No	9	23	20	12	29	3	23	9	22	10	32	0	17	15	31	1	30	2	31	1	31	1
	Si	18	7	19	6	25	0	12	13	23	2	0	25	6	19	25	0	25	0	25	0	25	0
Implement	No	5	18	10	13	20	3	17	6	18	5	17	6	23	0	22	1	21	2	23	0	22	1
	Si	22	12	29	5	34	0	18	16	27	7	15	19	0	34	34	0	34	0	33	1	34	0
Gest. Aud externa	No	27	29	38	18	53	3	35	21	44	12	31	25	22	34	56	0	54	2	55	1	55	1
	Si	0	1	1	0	1	0	0	1	1	0	1	0	1	0	0	1	1	0	1	0	1	0
Testing	No	26	29	38	17	52	3	33	22	43	12	30	25	21	34	54	1	55	0	54	1	54	1
	Si	1	1	1	1	2	0	2	0	2	0	2	0	2	0	2	0	0	2	2	0	2	0
Qual assur-	No	26	30	39	17	53	3	34	22	44	12	31	25	23	33	55	1	54	2	56	0	55	1
	Si	1	0	0	1	1	0	1	0	1	0	1	0	0	1	1	0	1	0	0	1	1	0
Consult	No	27	29	38	18	53	3	34	22	44	12	31	25	22	34	55	1	54	2	55	1	56	0
	Si	0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1

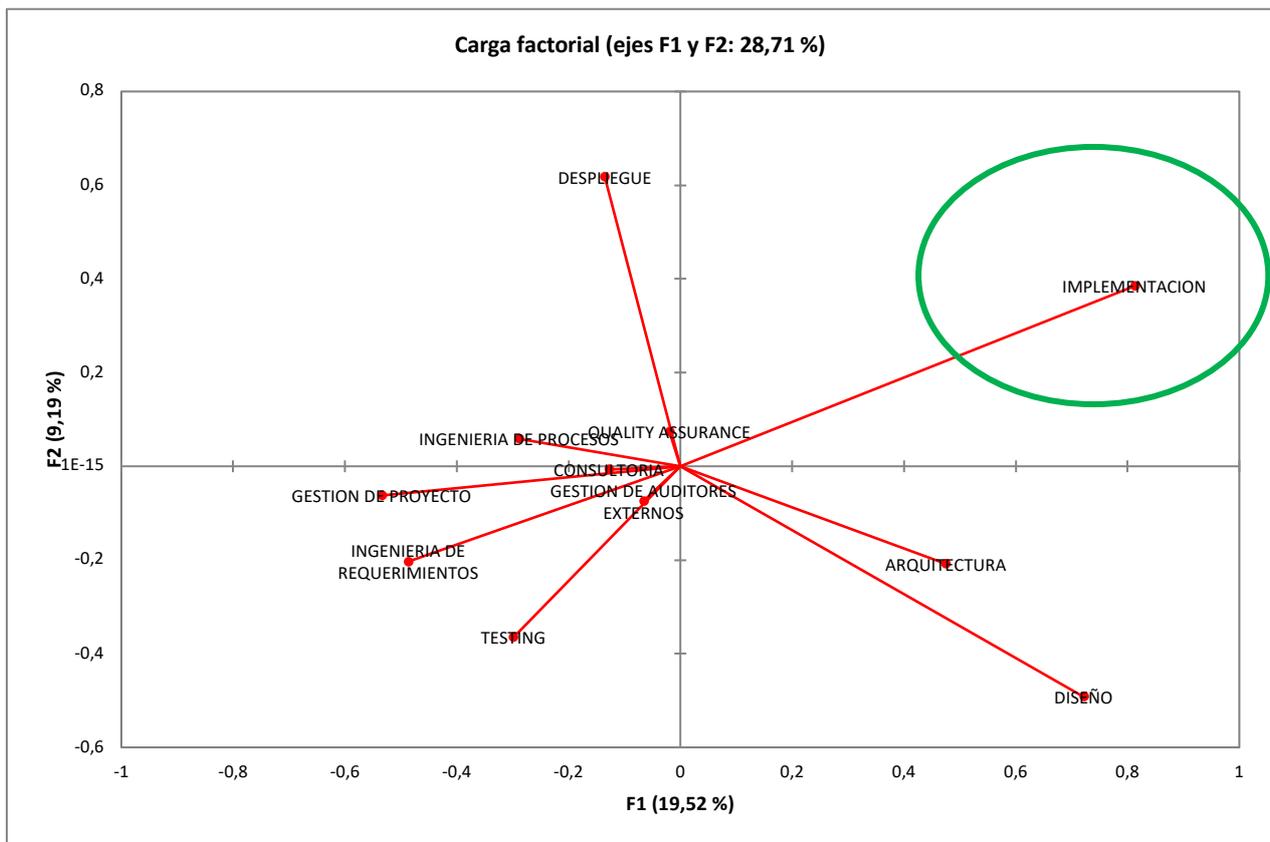


Gráfico 44. Tipologías Análisis factorial para la variable tareas que se desarrollan
Fuente: Elaboración propia

Si se observa el Gráfico 44 se puede inferir que, en relación a las tareas que quienes se desempeñan en el sector desarrollan, existen cuatro perfiles bien diferenciados a) quienes se desempeñan en el ámbito del Diseño y la Arquitectura, por un lado, quienes se desempeñan en el ámbito de la Implementación y por el otro, aquellos que lo hacen en el área de Ingeniería de Procesos, Gestión de proyectos e Ingeniería de Requerimientos. Por otro lado, quienes se desempeñan en implementación y aquellos que lo hacen en el área de despliegue. El resto de las actividades, conforman un conglomerado, en los que por tratarse de tareas con menor peso no aparecen perfiles tan fuertemente delimitados.

Variable Aspectos que conspiran con la obtención de un producto software de calidad

En la Tabla 20 y Gráfico 45 se pueden visualizar los valores obtenidos para cada una de las subvariables que en la que se desagrega la variable aspectos que por ausencia o déficit conspiran contra la elaboración de software de calidad.

Al respecto se observa que la falta de comunicación (58,92%), el involucramiento del cliente (55,35), falta de gestión e insuficiente información al asumir el compromiso (ambas con el 48,21%) son los factores

que los encuestados han considerado más determinantes en relación con la obtención de un producto de calidad.

Por el contrario, la falta o insuficiente adaptación de metodologías o procesos a utilizar (100%), la subestimación del producto a construir (98,21%), las relaciones interpersonales deficitarias (94,64), y la ausencia de un liderazgo efectivo (71,49) aun conocimiento técnico no adecuado (69,64%) no resultan factores determinantes para alcanzar calidad en los productos y servicios que se desarrollan.

Tabla 20. Estadísticas Descriptivas variable aspectos que por su ausencia o déficit conspiran contra software de calidad

Fuente. Elaboración propia

Sub-Variable	Categorías	Frecuencias	%
gestión	No	29	51,786
	Si	27	48,214
comunicación,	No	23	41,071
	Si	33	58,929
liderazgo	No	40	71,429
	Si	16	28,571
conocimiento técnico	No	39	69,643
	Si	17	30,357
nivel de involucramiento del Cliente	No	25	44,643
	Si	31	55,357
relaciones interpersonales	No	53	94,643
	Si	3	5,357
estimaciones	No	37	66,071
	Si	19	33,929
Información al asumir compromisos	No	29	51,786
	Si	27	48,214
Subestimación del producto a construir	No	55	98,214
	Si	1	1,786
adaptación de la metodología o proceso a utilizar	No	56	100
	Si	0	0

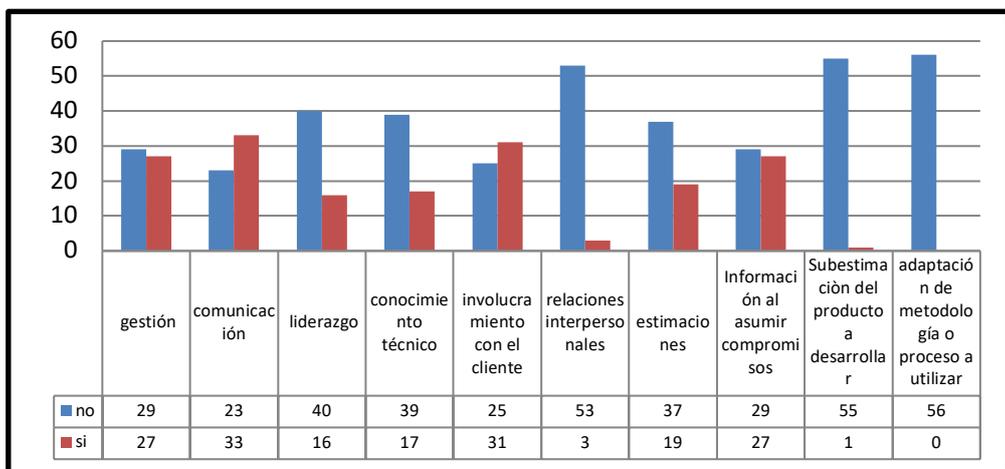


Gráfico 45. Aspectos que conspiran contra la calidad de los productos y o servicios de software

Fuente: Elaboración propia

Tabla 21. Factores que por su déficit o ausencia conspiran con la calidad del producto o servicio

		gestión-		comunicación		liderazgo		conocimiento técnico		nivel de involucramiento del Cliente		relaciones interpersonales		Malas estimaciones-		Información al asumir compromisos		Subestimación del producto a construir		metodología o proceso a utilizar
		no	si	no	si	no	si	no	si	no	si	no	si	no	si	no	si	no	si	no
gestión	Mo	29	0	8	21	20	9	21	8	12	17	26	3	20	9	12	17	29	0	29
	Si	0	27	15	12	20	7	18	9	13	14	27	0	17	10	17	10	26	1	27
comunicación	No	8	15	23	0	19	4	14	9	8	15	23	0	13	10	9	14	23	0	23
	Si	21	12	0	33	21	12	25	8	17	16	30	3	24	9	20	13	32	1	33
liderazgo	No	20	20	19	21	40	0	29	11	16	24	38	2	23	17	16	24	40	0	40
	Si	9	7	4	12	0	16	10	6	9	7	15	1	14	2	13	3	15	1	16
conocimiento técnico	No	21	18	14	25	29	10	39	0	18	21	36	3	23	16	15	24	39	0	39
	Si	8	9	9	8	11	6	0	17	7	10	17	0	14	3	14	3	16	1	17
nivel de involucramiento del Cliente	No	12	13	8	17	16	9	18	7	25	0	22	3	11	14	14	11	25	0	25
	Si	17	14	15	16	24	7	21	10	0	31	31	0	26	5	15	16	30	1	31
Relaciones interpersonales	No	26	27	23	30	38	15	36	17	22	31	53	0	34	19	27	26	52	1	53
	Si	0	3	3	0	2	1	3	0	3	0	0	3	3	0	2	1	3	0	3
Estimación incorrecta	No	20	17	13	24	23	14	23	14	11	26	34	3	37	0	20	17	36	1	37
	Si	9	10	10	9	17	2	16	3	14	5	19	0	0	19	9	10	19	0	19
información al asumir el compromiso	no	12	17	9	20	16	13	15	14	14	15	27	2	20	9	29	0	28	1	29
	Si	17	10	14	13	24	3	24	3	11	16	26	1	17	10	0	27	27	0	27
Subestimación del producto a construir	No	29	26	23	32	40	15	39	16	25	30	52	3	36	19	28	27	55	0	55
	Si	0	1	0	1	0	1	0	1	0	1	1	0	1	0	1	0	0	1	1
metodología o proceso a utilizar	No	29	27	23	33	40	16	39	17	25	31	53	3	37	19	29	27	55	1	56

A partir de los datos obtenidos en la Tabla 21, se desarrolla el análisis factorial que permite obtener tipologías para la variable factores que conspiran o dificultan alcanzar productos software de calidad.

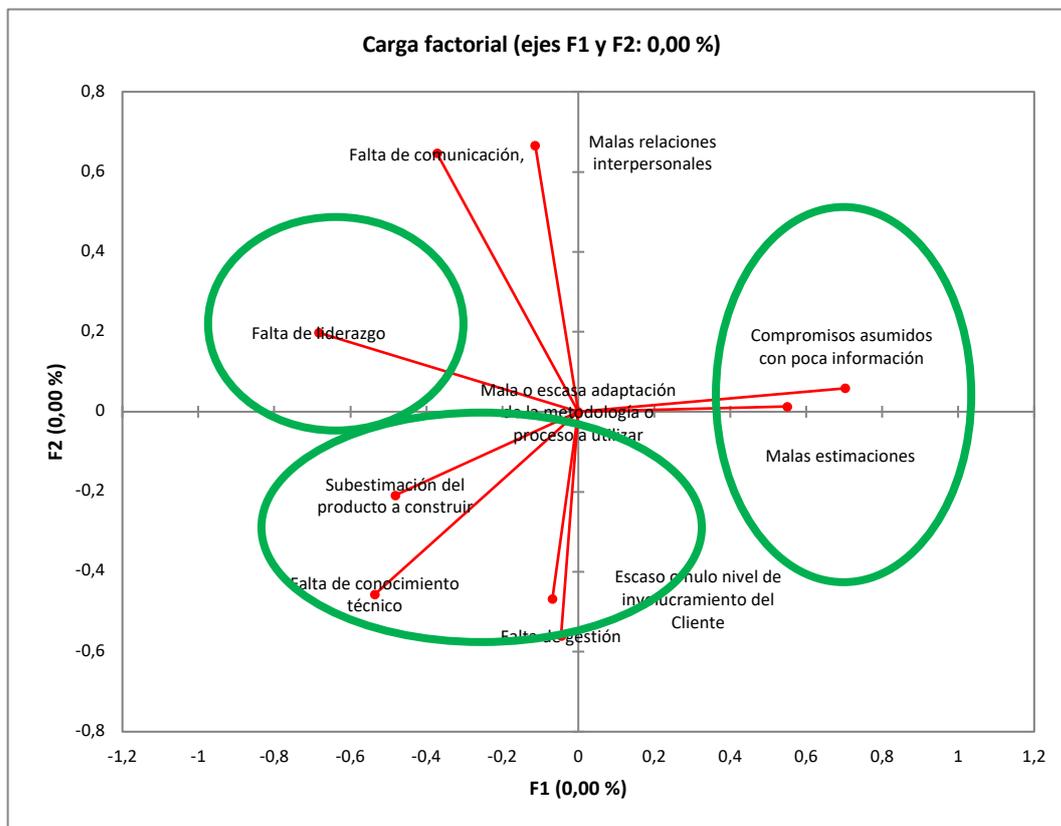


Gráfico 46. Tipología Análisis factorial para la variable factores que contribuyen a alcanzar productos de calidad
Fuente: Elaboración propia

Si observamos el Gráfico 46 correspondiente al análisis factorial, encontramos las siguientes tipologías en función de las respuestas en torno a la variable que describe cuales son aquellos factores que contribuyen a que los equipos de trabajo no alcancen calidad en los productos que desarrollan. Es así como por un lado estarían aquellos aspectos que implican asumir compromisos con poca información o malas estimaciones en general respecto de la actividad. Por otro lado problemas con el liderazgo. Otro grupo de factores son la falta de comunicación y las relaciones interpersonales inadecuadas. Por último el resto de los factores se agrupan en otra tipología de problemas tales como la subestimación del producto a construir, la falta de conocimiento técnico, la falta de gestión entre otros.

CAPITULO VII

Discusión y Presentación del Modelo

Principales Conclusiones

En virtud de los resultados obtenidos, se plantean las siguientes conclusiones teniendo en cuenta para ello las hipótesis que dieron lugar a la investigación.

Según se plantea en la primera hipótesis, La utilización de metodologías ágiles en el proceso de construcción de software, mejora la calidad del producto resultante.

Tal como se expone en el Capítulo III, el propósito principal de estas metodologías es lograr un equilibrio entre los procesos disciplinados que proponen una excesiva carga de trabajo y burocracia, y la no existencia de proceso, que lleva a desarrollar software de manera caótica.

La alternativa que ofrecen las metodologías ágiles propone un cambio radical que hace énfasis en un replanteo de aspectos culturales, comunicacionales y sociales, que son los principales causantes de fracasos en los proyectos de desarrollo de software.

La propuesta más desafiante del manifiesto ágil tiene que ver con evaluar cada acción, cada decisión, cada elección con el propósito de entregar “valor de negocio” a quién solicita el software, que a su vez se siente parte del proyecto.

Aspectos relevantes de los resultados que fueron seleccionados y se resumen con el fin de direccionar un una propuesta de optimización (estándar) para los equipos de desarrollo software:

Del análisis multivariado que se realiza del punto de interrogación en el cuestionario: ¿En los equipos de desarrollo de software en los cuáles trabajó o trabaja actualmente, qué competencias técnicas son requeridas? Se desprende que los aspectos; Gestión ágil de proyectos; Gestión ágil de requerimientos, que son propios de las metodologías ágiles, son de los más requeridos y que el más requerido es el vinculado a la Programación o Desarrollo, en este caso, para todos los métodos, tanto los definidos, como los ágiles. Por lo tanto, y a partir de los resultados obtenidos en relación con los aspectos solicitados, se puede inferir que trabajar con estas metodologías, mejorarían la calidad de los productos.

También podemos mencionar los resultados vinculados a las variables que se consulta: ¿En los equipos de desarrollo de software en los cuáles trabajó o trabaja actualmente, qué competencias conductuales son requeridas? y según su criterio, ¿cuál de los siguientes aspectos de la relación vincular entre los involucrados en el desarrollo de un producto de software, es necesario mejorar? A partir de las

respuestas se advierte que las competencias más solicitadas, están íntimamente vinculadas a las requeridas por estas metodologías

En la tabla 11 y gráfico 30 se pueden visualizar los valores obtenidos para cada una de las subvariables que en la que se desagrega la variable aspectos que por ausencia o déficit conspiran contra la elaboración de software de calidad. A partir de estos resultados, se advierte que las variables que conspiran con la calidad del software son las que denotan una ausencia de metodologías ágiles.

Por otra parte, y de acuerdo con los datos que se presentan en el gráfico 39 , los resultados dan cuenta que las metodologías ágiles colaboran en la obtención de productos software de calidad.

No obstante, se puede inferir que hay que trabajar mucho en instalarlas, dado que aunque se reconozca la necesidad de uso de estas o de sus propiedades, para desarrollar software de calidad, podemos apreciar, como se ve en el grafico 21, que su utilización es aún muy baja.

Se ha planteado en la segunda hipótesis de esta investigación que un software que responde a las necesidades de los usuarios satisfactoriamente (un software con calidad), está relacionado con las competencias de los stakeholders, especialmente de los desarrolladores; a su vez en la tercera hipótesis, se sostiene que la comunicación entre los stakeholders es un factor que interviene de forma crítica en el desarrollo del software.

Considerando los resultados de la Tabla 11 y de Gráfico 45 del análisis factorial se advierte que los factores determinantes que los encuestados observan respecto a los aspectos que conspiran en la obtención de un producto software de calidad son: falta de comunicación (58,92%), involucramiento del cliente (55,35), falta de gestión e insuficiente información al asumir el compromiso (ambas con el 48,21%);

Esto pone de manifiesto la ausencia de utilización de una de las soluciones a estos problemas traídas por las metodologías empíricas, o sea los denominados métodos ágiles.

Respecto la falta de comunicación y al involucramiento del cliente, es de destacar que el manifiesto ágil promueve una solución para aquellos factores que atentan contra la calidad del software, que gira en torno de privilegiar lo que es de valor para el cliente, es decir, entregas frecuentes de software funcionando. Propone un ambiente de colaboración con el cliente/usuario y entre los miembros del equipo y una actitud de aceptación frente a los cambios que surjan durante el desarrollo del producto de software, rescatando valores como la gestión y la comunicación. Creando la figura del Product Owner: el dueño del producto, la voz principal del cliente, quien asume la responsabilidad por su construcción y por maximizar el valor de negocio, propiciando el involucramiento del cliente. Sin embargo, del resultado de la encuesta se desprende que solo un 8 % de la población encuestada utiliza métodos ágiles, por lo tanto, mayoritariamente no se benefician de las bondades de estas metodologías.

En el capítulo III, mencionamos que el desarrollo de un Esquema Conceptual Referencial y Operativo común a los miembros del grupo permite el incremento de la comunicación intragrupal ya que, de acuerdo con la teoría de la información, lo que permite que el receptor comprenda el mensaje emitido por el transmisor, a través de operaciones de codificación y decodificación, es una semejanza de esquemas referenciales. En este proceso de comunicación y aprendizaje observamos que el grupo sigue un itinerario que va del lenguaje común al lenguaje científico.

Respecto a la Insuficiente información para asumir el compromiso considerando que la definición del dominio del problema y la determinación de los requerimientos son algunas de las actividades encuadradas dentro de la Especificación del Software y es justamente aquí donde se concentran los mayores problemas a la hora de construir software.

En el planteo de la cuarta hipótesis, se señala que la trama vincular en el equipo de desarrollo, incluyendo al dueño del producto (Product Owner del Scrum) y a los usuarios directos, es un factor determinante de la calidad del producto a desarrollar.

En el análisis descriptivo de los resultados, al interrogante del cuestionario sobre los aspectos de la relación vincular entre los involucrados en el desarrollo de un producto de software que es necesario mejorar; se observa de los datos que surgen del Gráfico 28, que la comunicación (36%) es el aspecto de la relación vincular que aparece como central para el desarrollo de software de calidad. Dato que coincide con los resultados obtenidos en el apartado anterior. Estos resultados permiten inferir la importancia de la comunicación en los procesos que este trabajo explora.

Pichón Rivière sostiene que los procesos de codificación y decodificación de señales pertenecen a esquemas referenciales individuales y de los grupos a través de los que se hace posible, según sea el funcionamiento y la estructura de estos esquemas, configurar situaciones de entendimiento o mal entendimiento. En última instancia, la comunicación grupal es posible por la existencia de un ECRO grupal.

Respecto a la comunicación, Pichón Riviere plantea: "*Comunicación es todo proceso de interacción social por medio de símbolos y sistemas de mensajes. Incluye todo proceso en el cual la conducta de un ser humano actúa como estímulo de la conducta de otro ser humano. Puede ser verbal, o no verbal, interindividual o intergrupal*" (Riviere, 1997)

Por otro lado, la cooperación, tal como se señaló, se refiere a la elección por parte de los miembros del equipo de desarrollo de roles diferenciados y complementarios para el logro de la tarea, sin que esto signifique substituir ni apartar a ningún otro miembro. Este aspecto en el estudio fue ponderado con un 27%.

La cooperación es la capacidad de articularse con otros para realizar una tarea; es la contribución, aún silenciosa, a la tarea grupal, sobre la base de la existencia de roles diferenciados. La pertinencia es la capacidad de centrarse en la tarea prescrita es decir el involucramiento.

Respecto al eje del compromiso entendido como competencia de una persona para tomar consciencia de la importancia que existe en cumplir con algo acordado anteriormente, considerándolo como una relación no lineal sino dialéctica entre los sujetos y el involucramiento de actores, es una respuesta al problema de la legitimidad de las decisiones, garantizando los plazos de ejecución de los trabajos; en este caso obtuvieron el 18 % y 17 % respectivamente.

Teniendo en cuenta los análisis estadísticos realizados, se pueden presentar algunas otras conclusiones que no estuvieron previstas en el planteo hipotético original. Las siguientes conclusiones son resultado del Análisis Bivariado por medio del cual se cruzaron dos o más preguntas de la encuesta, la riqueza de este cálculo permite, extraer las siguientes conclusiones.

En relación con las variables ¿Cuáles son las tres principales tareas que usted realiza en su trabajo? y Sexo:

Este resultado permite afirmar que no existe dependencia entre ambas variables, por lo que las tareas que se realizan en los ámbitos laborales de la Informática no están relacionados al sexo de quien las realizan. Esa conclusión es correcta, no obstante sabemos que a mediados de los 90 la población femenina disminuyó fuertemente de ser un 50% o más de la fuerza laboral, hoy es un solo un 20%, porcentaje aún menor al que nos muestran los resultados de la encuesta.

Por otro lado cuando indagamos sobre los motivos, por medio de un focus group, con un panel de 6 mujeres profesionales, obtenemos como resultado que los permanentes viajes, el horario extendido en el trabajo, las presiones en los ámbitos de trabajo, entre otros son los motivos de la disminución de esta población, no solo por la elección de no trabajar siendo de la profesión, sino ya llega a la no elección de este tipo de carreras. Cabe agregar que se están realizando acciones, desde las organizaciones educativas, gubernamentales y varias ONGs para revertir esta situación, entre otro el motivo es por la gran necesidad que existe de incorporar esa parte de la población a la industria del software.

Para las variables Principales tareas que se realizan en el trabajo y si en el ámbito del equipo de desarrollo en el que trabaja, utiliza metodologías basadas en procesos definidos, si es así ¿cuáles?

En este caso, y a partir de los datos obtenidos en el presente estudio, se puede afirmar que ambas variables, tareas que se realizan y metodologías basadas en procesos definidos, tienen un funcionamiento independiente una de la otra.

Por otra parte, al relacionar las variables principales que usted realiza en su trabajo? y competencias conductuales son requeridas en los equipos en los que se desempeña

El resultado de este cruce de datos denota la importancia que para el desarrollo de las diferentes tareas tienen las competencias conductuales. Una diferenciación de los perfiles demandados está relacionada con dichas competencias y por lo tanto le son requeridas en función de cada perfil.

Las competencias más demandadas dentro de las conductuales son la resolución de problemas, el trabajo en equipo y la autoorganización, competencias estas más requeridas para las tareas de gestión de proyectos, ingeniería de requerimientos e implementación.

Estos procesos, son por los que pasa un grupo de trabajo centrado en la tarea, y con una espiral dialéctica de mutua realimentación: mutua representación interna y en constantes de espacio-tiempo (en la actualidad con peculiaridades propias de la era digital) características que son sustantivas para comprender las relaciones y vínculos interpersonales, la relación con la tarea (involucramiento y compromiso) papel del líder y conductor del grupo, los miedos y resistencias que pueden surgir en un proceso creativo que constituye el desarrollo (como el aprendizaje espiralado), la adaptación activa a la realidad y la participación de stakeholders durante el proceso de desarrollo.

Otro aspecto que se indagó a través del análisis bivariado, fueron las siguientes variables competencias conductuales competencias técnicas requeridas.

En este caso, se realizó el cruzamiento de dos tipos de competencias que parecen antagónicas no obstante el resultado, de este cruce pareciera demostrar que el conjunto intersección está vacío entre ambos tipos de competencias, los aspectos técnicos y conductuales no tienen relación entre sí. Empero como el desarrollo de software es una actividad que se realiza principalmente en equipo, o sea el grupo está centrado en la tarea de crear y desarrollar software; cada una las competencias técnicas involucradas en esa creación son importantes y diferentes, para la ejecución de todas ellas es necesaria la presencia de todas las competencias conductuales analizadas destacando muy especialmente las relacionadas con trabajo en equipo autoorganización y resolución de problemas complejos, las que deben estar presentes en forma constante y permanente para que el equipo de trabajo alcance las metas propuestas.

Respecto a las competencias técnicas específicamente, los resultados permiten afirmar que dentro de las competencias técnicas la que más se requiere es la vinculada a la programación, por ser de todas las actividades afectadas a la construcción de software la única no prescindible. Si las demás no se despliegan trae como consecuencia software de baja calidad sin la confiabilidad requerida con una tasa de fallas muy alta.

Desde el inicio del desarrollo de software, la función principal era la programación, el resto se desarrollaban en forma no sistemática y muchas veces intuitiva podemos mencionar los requerimientos, análisis, diseño, pruebas, entre otras. El surgimiento de las metodologías para el desarrollo de software incorporó estas actividades a medida que se detectaba que en la cadena de valor los eslabones más débiles eran los que generaban dificultades en la creación de tan particular producto, no obstante el surgimiento de estas etapas, no fueron incorporados rápidamente en los procesos de desarrollo. Fue necesario introducir la gestión del proyectos tanto en formas de procesos definidos como en los procesos empíricos para obtener resultados esperados y un software confiable y de calidad.

Por último si se analizan los resultados para las variables competencia conductuales requeridas y aspectos de la relación vincular que es necesario mejorar, se observa que la relación entre las competencias conductuales necesarias y los aspectos de la relación vincular, son mínimos. Pero sin embargo, se advierte que hay evidencia objetiva, que los aspectos a mejorar más frecuentes son en primer lugar la comunicación muy apreciablemente, también la cooperación y luego el compromiso o pertenencia, como así también el involucramiento o pertinencia.

Si los confrontamos con las competencias conductuales se puede reparar que las más requeridas son el trabajo en equipo, la autoorganización y la resolución de problemas complejos; las cuales no son posibles sin la manifestación de las otras.

Propuesta de un modelo de trabajo conceptual referencial para los equipos de desarrollo de software

Se propone un modelo de trabajo que define un conjunto de conceptos que son aplicables a áreas de ingeniería de procesos y en particular a todos los proyectos de desarrollo de software, con el objetivo de ayudar a los profesionales a evaluar la salud del grupo como equipo de trabajo, comprometido en la creación de software y en mejorar su forma de vincularse entre todos los involucrados directos e indirectos.

El modelo busca reforzar aspectos culturales que conlleven a mejorar las estructuras organizacionales, dado que a pesar del hecho que, según Larman (Larman, 2017) plantea en uno de sus postulados conocidos como *las leyes de Larman*, “la cultura sigue a la estructura”, es decir el mecanismo de funcionamiento del equipo es un reflejo del mecanismo de funcionamiento de la organización. La propuesta apunta a, en un sentido inverso, trabajar sobre los equipos para provocar un cambio y una mejora que se traslade a las organizaciones (botton up).

Analizando el comportamiento de los grupos que adhirieron a prácticas ágiles, en su mayoría se observan aspectos “débiles” en su adopción. Esto obedece a una raíz común, la falta de comprensión de las prácticas ágiles, especialmente las relacionadas con valores y cimientos de la cultura ágil; particularmente con aspectos de la trama vincular, ya que la misma aparece como superficial, subestimada, de forma tal que, cuando aparecen algunos problemas, tanto el equipo como los administradores terminan tomando decisiones “no-ágiles”. Es preciso considerar que en estas metodologías la producción depende del trabajo en equipo y no de las individualidades; por lo que banalizar la trama vincular conspira contra la productividad.

Este modelo de trabajo se manifiesta en la generación de heurísticas¹⁹ y de indicadores clave de vínculos. Las heurísticas está agrupadas en ejes temáticos los cuales se enumeran a continuación:

¹⁹ eurísticas son metodologías que encuentran la capacidad para realizar, de forma inmediata, cambios positivos para sus fines. También, se dice que la heurística es un rasgo netamente humano; desde este punto de vista, se define como el arte del descubrimiento y de la invención o de la resolución de problemas, mediante la creatividad y el pensamiento lateral (Bono, 1967).

comunicación entre los stakeholders, clima laboral, tele entre los miembros del grupo, pertenencia, pertinencia y cooperación.

Así mismo, los indicadores clave de vínculos para un equipo de proyecto de desarrollo de software, respetan los mismos ejes temáticos de las heurísticas y constituyen la herramienta de evaluación de su grado de presencia en los equipos de trabajo.

El modelo propuesto puede aplicarse al factor humano en procesos de ingeniería flexible, y en particular a los equipos de desarrollo de software. El propósito es obtener beneficios para individuos, equipos y organizaciones.

A continuación se presentan las heurísticas, organizadas por ejes temáticos enfocadas a los vínculos que se establecen en los equipos de trabajo:

- **Comunicación entre los stakeholders:**

- **Visión compartida:**

El éxito en el logro de los objetivos de un proyecto está fuertemente relacionado con el grado en el que los stakeholders (involucrados internos y externos) puedan construir una visión integrada y priorizada de cuáles son esos objetivos. La base de construcción de esa visión compartida es la comunicación. Alcanzar una comunicación efectiva entre personas con perfiles heterogéneos, que es lo que ocurre en los proyectos con la presencia de todos los interesados es más difícil aún. Se debe trabajar en pos de hacer visible la visión de cada una de las personas para construir desde allí la visión colectiva.

- **Reuniones como herramienta de creación de valor:**

La comunicación cara a cara es el mejor medio de comunicación, para lograr el propósito pretendido al convocar a una reunión debemos tener un objetivo claro y alcanzable, elegir a los participantes relevantes, definir un “timebox” (duración de la reunión) y respetarlo, crear una agenda, distribuirla y cumplirla, obtener feedback de la reunión al finalizar.

- **Clima de trabajo:**

- **Gestión del tiempo**

Uno de los grandes desafíos que enfrentan las personas que trabajan con tareas intelectuales como es el caso específico del desarrollo de software, es el uso que hacen del tiempo. Para lograr un buen desempeño y aumentar la satisfacción personal y profesional es de utilidad tener en cuenta estos aspectos que ayudan con la administración del tiempo: tener claro el objetivo, priorizar el trabajo que se requiere para alcanzar el objetivo, organizar el espacio de trabajo, procurando disponer de todos los recursos necesarios, alternar tipos de actividades diferentes.

- **Foco en la tarea**

Un corolario de la práctica anterior, que a su vez aporta a su logro, es la habilidad de concentrar los esfuerzos en una tarea a la vez. Evitar el solapamiento de tareas y la multitarea que genera un consumo de tiempo adicional en el cambio entre una tarea y otra (tiempo de seteo). Durante la realización de esa tarea es recomendable minimizar las interrupciones, evitar distractores y dimensionar de forma realista el esfuerzo requerido para cumplirla.
- **Uso de espacios compartidos:** los equipos trabajan mejor cuando comparten espacios de trabajo, estos espacios deben diseñarse para facilitar el desarrollo de la tele, la pertenencia, la pertinencia y la cooperación. Espacios de trabajo que permitan visualizar y compartir el trabajo.
- **Tele entre los miembros del grupo** (involucrados internos): este eje temático se aborda desde las siguientes prácticas:
 - **Respeto por el grupo:**

Uno de los valores que permite mantener un ambiente de trabajo propicio para el desarrollo y satisfacción de las personas que conforman el equipo es el respeto. El respeto por cada persona como individuo y el respeto por el equipo. El respeto crea un ambiente de trabajo cordial y seguro que sienta las bases para que cada miembro del grupo pueda desarrollarse, expresarse, aceptar las fortalezas y limitaciones de cada persona y las del equipo.
 - **Capacidad de Expresión:**

La comunicación del grupo se basa en la capacidad de expresión de cada una de las personas que lo integran. Es importante desarrollar el potencial de expresión, especialmente la comunicación verbal de cada uno de los involucrados, para generar sinergia de la expresión del equipo.
 - **Construcción de Equipo:**

El grupo de personas que comienzan a trabajar en un proyecto deben aprender a conformarse como equipo. Para ello se sugiere desarrollar la capacidad de autoorganización de los equipos. Este concepto es esencial para lograr constituir una cultura interna del equipo y un estilo de trabajo.
- **Pertenencia:**
 - **Objetivos compartidos**

Los objetivos derivados de la visión compartida constituyen los logros del equipo. Estos objetivos deben ser claros, alcanzables, acordarse y priorizarse entre todos los miembros del equipo y estar visibles.

- **Compromiso**
El compromiso está directamente relacionado con el logro de los objetivos compartidos y se pone de manifiesto cuando se logra que estos objetivos se conviertan en los objetivos personales de cada uno de los integrantes; el compromiso también se pone de manifiesto siendo honesto y generoso con la información.

- **Conducta activa**
La proactividad como actitud de trabajo. Se espera que cada persona tome el control y decida qué hacer frente a cada situación, esto implica la toma de iniciativa en el desarrollo de acciones creativas y audaces para generar mejoras, haciendo prevalecer la libertad de elección responsable y la transformación de ideas en acciones. La conducta activa se manifiesta en actuar buscando oportunidades, tratando de anticiparse, prevenir y resolver problemas y no limitarse a ser sólo reactivo.

- **Pertinencia:** para el eje temático de pertinencia se identifican las siguientes prácticas:
 - **Identificación con el equipo**
Cada uno de los miembros del equipo siente que forma parte del grupo y reconoce el valor de su presencia en la conformación del equipo. Cada miembro tiene mutua representación interna de los otros integrantes.

 - **Equipo Involucrado:**

Cada miembro del equipo comprende la incidencia de su participación y el impacto que causa su accionar en el equipo.

 - **Equipo Integrado:**

Un equipo integrado logra un vínculo que construye comunidad y esa “comunidad” facilita que el trabajo fluya.

- **Cooperación:** el eje temático de cooperación es el resultado de una estrategia de trabajo conjunto que desarrolle el pensamiento colaborativo, para este eje se destacan las siguientes prácticas:
 - **Colaboración**
La comunicación implica colaboración, el trabajo colaborativo implica cooperación mutua y es el vehículo que viabiliza el desarrollo hacia la creatividad y la innovación. Cada uno de los miembros del equipo debe poder equilibrar sus intereses y el cumplimiento de sus objetivos con el logro de las metas del equipo.

- o **Asistencia/ Acompañamiento**

Cada miembro del equipo debe aprender a brindar asistencia efectiva. Esto implica la capacidad de discernir en cada situación, que colaboración es la que debe ofrecer a los otros miembros. Cada miembro del equipo debe aprender a ofrecer y realizar tareas de acompañamiento realizando la tarea de forma conjunta.

- o **Solicitud de asistencia**

Como miembro de un equipo cada integrante deberá desarrollar tanto la habilidad de ofrecer asistencia como la de pedirla. En este caso, la práctica se refiere a la capacidad de identificar el momento y la forma adecuadas para pedir colaboración a otro miembro del equipo.

Los *indicadores clave de vínculos*, son la herramienta propuesta para ayudar a los equipos a evaluar su nivel de internalización e institucionalización de las heurísticas y como corolario el nivel de madurez de los vínculos que constituyen su esquema conceptual-referencial.

Para hacer un análisis cuantitativo, a cada indicador clave de vínculos presentado, se le asignará un valor en función de la siguiente escala:

0. **Nulo:** para indicar ausencia del eje temático evaluado.
1. **Impedido:** equipo con fuerte dificultad de vincularse.
2. **En transición:** equipo que está trabajando los vínculos.
3. **Sostenible:** el equipo logra establecer vínculos y mantenerlos.
4. **Ideal/Óptimo:** el equipo logra sostener los vínculos y mejorarlos.

Se propone una rúbrica²⁰ que permita evaluar el índice en cada eje temático. La rúbrica es un instrumento que sirve para evaluar el eje en función de los indicadores de vínculos, teniendo en cuenta para ello un conjunto de criterios que han sido acordados en forma previa a su utilización por parte del equipo. En la rúbrica, se define el valor alcanzado por el grupo en cada eje.

En la tabla 22 se muestra la rúbrica de referencia, para la autoevaluación del equipo.

²⁰ Rúbrica: Las rúbricas son guías precisas que valoran los aprendizajes y productos realizados. (Gatica-Lara & Uribarren-Berrueta, 2012)

Indicador de vínculo → Eje temático ↓	0. Nulo	1. Impedido	2. En transición	3. Sostenible	4. Ideal/Óptimo
Comunicación entre los stakeholders	No construida	Comunicación en etapa inicial, generando las bases de una visión compartida, reuniones no efectivas	En proceso de construcción de la visión compartida, estableciendo canales de comunicación; las reuniones comienzan a ser efectivas	Construida entre los involucrados internos y externos. Las reuniones son una herramienta incorporada a las prácticas del equipo	Construida y validada entre los involucrados internos y externos, La comunicación fluye y aporta valor; reuniones efectivas
Clima de trabajo	Uso no eficiente del tiempo; no se logra el foco en la tarea, espacios de trabajo no compartidos efectivamente	El tiempo se gestiona con dificultad; el foco en la tarea no logra concentrar los esfuerzos	El espacio físico comienza a ser un ámbito de trabajo; la concentración en la tarea es una práctica, el tiempo comienza a gestionarse	Las tareas se desarrollan en forma continua, el espacio es compartido; el uso del tiempo permite alcanzar los objetivos	Los objetivos se alcanzan con productos de calidad; la tarea no tiene interrupciones y el espacio compartido es un lugar agradable y acogedor, apto para la tarea
Tele entre los miembros del grupo	Comunicación existente	La comunicación entre los miembros del equipo está mejorando; el grupo comienza a trabajar en forma mancomunada.	Los miembros del equipo comienzan a reconocer los valores, habilidades y capacidades; el equipo comienza a autoorganizarse	Los miembros del equipo construyen un ámbito de confianza; la comunicación es fluida y el trabajo es continuo y sostenido	Los integrantes del equipo creen que los miembros tiene las capacidades necesarias para el trabajo; el proceso dialéctico se da plenamente
Pertenencia	No hay compromiso	Fuerte dificultad para identificarse entre sí	Comienza a tener objetivos compartidos, va adquiriendo compromisos y mutua representación	Se logra una visión compartida, compromiso y actitud proactiva	Los logros de cada miembro son los logros del equipo, el objetivo es uno y la conducta es proactiva
Pertinencia	No hay involucramiento	Fuerte dificultad para apropiarse del grupo y del trabajo	Se identificarse con la tarea y hay una integración en marcha	Involucrado e integrado	Cada miembro es el equipo, logrando un vínculo consolidado
Cooperación	No hay colaboración	Dificultad para funcionar como grupo colaborativo	Colabora y se establecen relaciones de asistencia	El trabajo es colaborativo y de cooperación mutua, los integrantes se asisten y acompañan	Se logra creatividad e innovación; la asistencia y el acompañamiento es parte de la interrelaciones

Tabla 22: Rúbrica de referencia de indicadores de vínculo, para la autoevaluación del equipo en relación con los ejes temáticos. Fuente: Elaboración propia

En la tabla 23 se muestra un ejemplo de aplicación del modelo y los indicadores clave de vínculos para un equipo de desarrollo de software:

Indicador clave de vínculo → Eje temático ↓		0. Nulo	1. Impedido	2. En transición	3. Sostenible	4. Ideal /Óptimo
	Heurística					
Comunicación entre los stakeholders	Visión compartida		X			
	Reuniones como herramienta de creación de valor			X		
Clima de trabajo	Gestión del tiempo		X			
	Foco en la tarea				X	
	Uso de espacios compartidos				X	
Tele entre los miembros del grupo	Respeto por el grupo			X		
	Capacidad de Expresión			X		
	Construcción de Equipo			X		
Pertenencia	Objetivos compartidos		X			
	Compromiso		X			
	Conducta activa			X		
Pertinencia	Identificación con el equipo				X	
	Equipo Involucrado			X		
	Equipo Integrado			X		
Cooperación	Colaboración		X			
	Asistencia /Acompañamiento			X		
	Solicitud de asistencia		X			

Tabla 23: Rúbrica para los Indicadores clave de vínculos por eje temático y heurística.
Fuente: Elaboración propia

Conforme el equipo se acerque al nivel 4: ideal/óptimo, en cada uno de los ejes temáticos, mostrará un mayor nivel de madurez en sus vínculos.

Esto se refleja en los gráficos radiales 47 y 48, que se muestran a continuación. Los mismos están basados en los indicadores del ejemplo de la tabla 23.

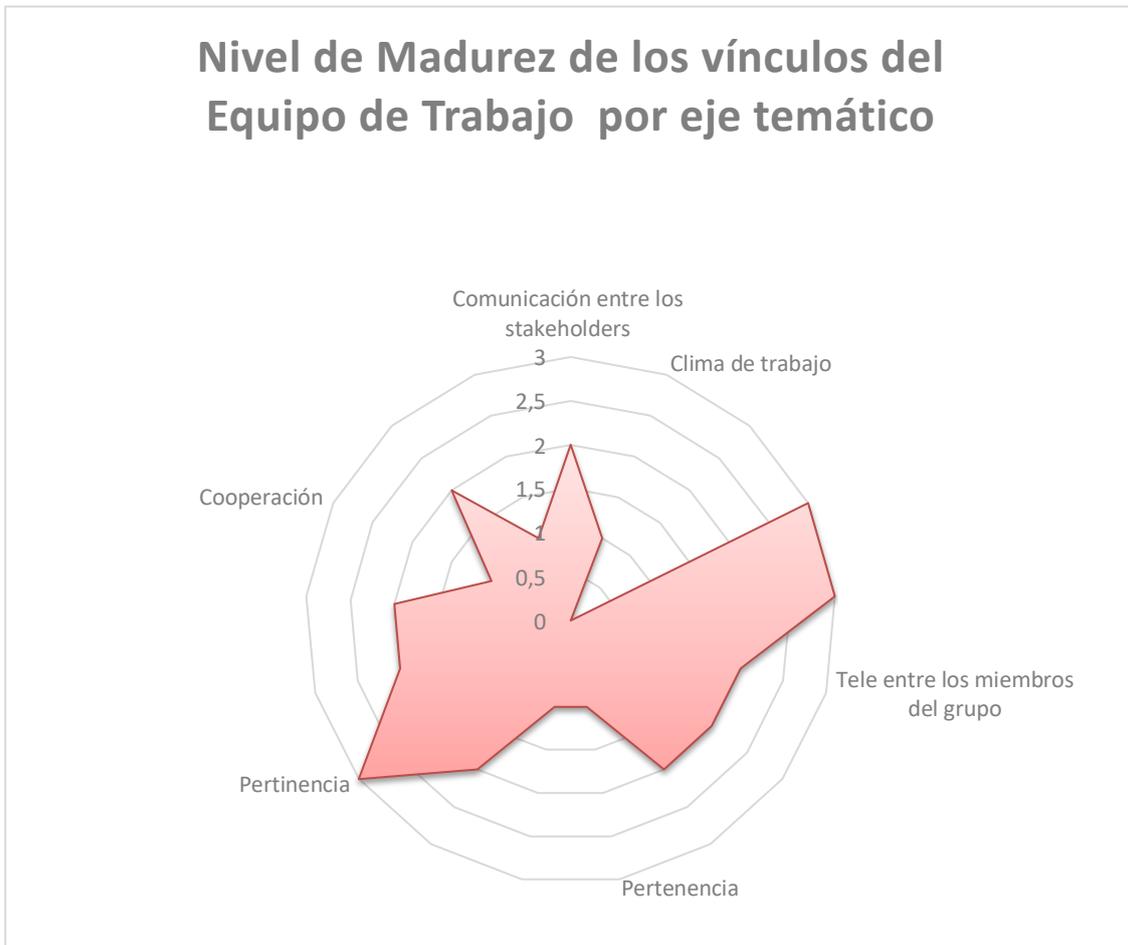


Gráfico 47: Gráfico radial que muestra el nivel de madurez de los vínculos del equipo de trabajo por eje temático para un Sprint Determinado.
Fuente: Elaboración propia

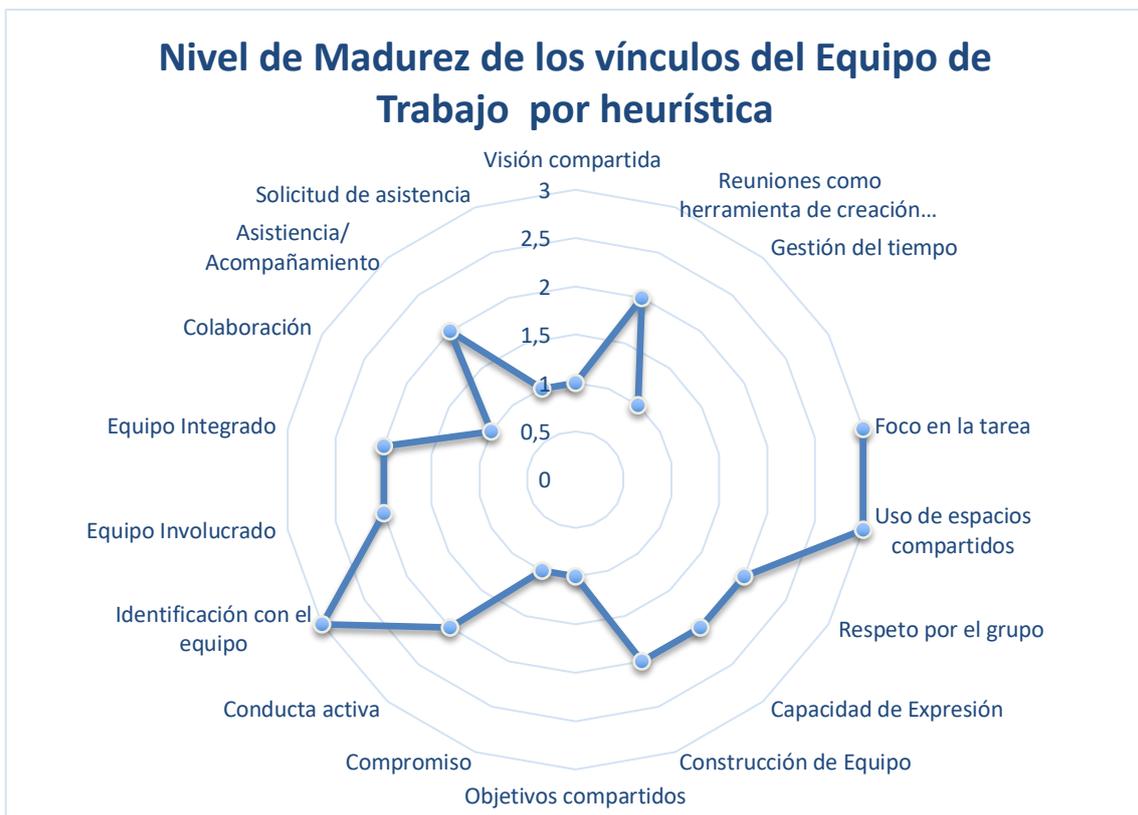


Gráfico 48: Gráfico radial que muestra el nivel de madurez de los vínculos del equipo de trabajo por heurística para un Sprint determinado
Fuente: Elaboración propia

Contexto de uso del modelo conceptual referencial

La propuesta recomienda la inclusión de las prácticas en forma permanente e integrada al trabajo de los equipos de desarrollo. Los indicadores clave de vínculo utilizados a partir de un framework ágil para la gestión de proyectos, en el contexto de la ceremonia que el equipo se reserva para inspeccionar y adaptar sus propias prácticas, conocida comúnmente como *Retrospectiva*, la cual parte de parte de Scrum.

Esta ceremonia es la materialización del último principio del manifiesto ágil que plantea que: “A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia”.

Tal como define Scrum, la retrospectiva es la última ceremonia que se realiza antes de cerrar el Sprint y brinda la oportunidad de reforzar y afianzar buenas prácticas y mejorar aquellas que no lo son tanto.

En este sentido y sustentado en la información obtenida del procesamiento de las encuestas, se observó que el foco de la Retrospectiva en la mayoría de los equipos está en mejorar aspectos relacionados con las

competencias técnicas, desplazando las competencias conductuales que son el núcleo del modelo conceptual referencial propuesto.

El modelo que se presenta propone fortalecer la ceremonia del Scrum denominada retrospectiva, con la inclusión del análisis de los ejes temáticos y las heurísticas propuestas; incorporando en ese momento la reflexión y medición de la trama vincular del grupo operativo.

Se propone la incorporación de una dinámica, en la cual, teniendo la tabla de referencia en forma visible y el framework de trabajo sobre el cual el grupo irá marcando el estado logrado al final de cada sprint. Esto será el insumo para los dos gráficos radiales que quedarán visibles en forma permanente, en las paredes del ámbito de trabajo.

Al sprint siguiente el grupo realizará una vez más, en la misma ceremonia (retrospectiva), la misma dinámica.

Al finalizar el proyecto, se propone una ceremonia final denominada “Retrospectiva Recargada” en la cual el grupo en un proceso de discernimiento evalúe la evolución en base al conjunto de frameworks y diagramas y pueda tomar acciones de mejora para futuros proyectos.

CAPITULO VII

Líneas a futuro

Habiendo finalizado el presente trabajo, consideramos que se abren nuevas líneas de trabajo con aspectos a explorar en futuros estudios y acciones de transferencia de los resultados obtenidos.

En particular se priorizan las siguientes líneas como prioritarias:

- Considerar no sólo las localidades maduras y extender el estudio a todo el país, en particular a las consideradas emergentes.
- Proponer un conjunto de recomendaciones para las currículas de las carreras superiores, tanto universitarias como terciarias, abordando también las capacidades requeridas para los técnicos.
- Ampliar el conjunto de recomendaciones para una formación continua.
- Instalar en la industria las necesidades detectadas pero aún implícitas en la misma, vinculadas a las competencias interpersonales de los grupos de trabajo.
- Aplicar el modelo que se propone y a través de un estudio longitudinal evaluar el impacto de su implementación en grupos de trabajo y en los resultados de la calidad del producto software alcanzados.

Bibliografía

- Agile Alliance. (2001). *The agile manifesto*. Obtenido de <http://www.agilealliance.org/the-alliance/the-agile-manifesto/>
- Barrows, H. (1996). Problem-Based Learning in Medicine and Beyond: A Brief Overview. (1996). *Bringing Problem-Based Learning to Higher Education: Theory and Practice.*, 3-12.
- Barrows, H. S. (1986). A Taxonomy of Problem-Based Learning Methods. *Medical Education*, 481-486.
- Basili, V. (1989). *The Software Business*. Obtenido de <http://www.cs.umd.edu/users/mvz/mswe609/book/chapter1.pdf>
- Basili, V. (1993). The Experimental Paradigm en Software Engineering. # 706 *Lecture Notes en computer software*. Springer-Verlag.
- Basili, V. S. (1986). *Experimentation in Software Engineering*. Recuperado de <http://www.dtic.mil/dtic/tr/fulltext/u2/a226830.pdf>.
- Beck, K. (2004). *Extreme Programming Explained: Embrace Change*. Boston: Addison Wesley.
- Bono, E. (1967). *The Use of Lateral Thinking*. Estados Unidos: Intl Center for Creative Thinking.
- Booch, G. (1991). *Object Oriented Design with Applications*. California: Benajmin / Cummings.
- Booch, G. R. (2005). *The Unified Modeling Language User guide* (Segunda ed.). Estados Unidos: Addison Wesley.
- Brooks, F. (1995). *The Mythical Man-Month: Essays on Software Engineering* (Aniversario ed.). Estados Unidos: Addison Wesley Logman.
- Cajal, M. (2018). *Mabel Cajal Turismo 2.0, Marketing Digital y Redes Sociales*. Recuperado el 2018, de <https://www.mabelcajal.com/2014/12/millennials-generacion-x-baby-boomers-como-se-comportan-online.html/>
- Davis, A. (1993). *Software Requirements, Objects, Functions & States*. Prentice Hall.
- Dirección de Investigación y Desarrollo Educativo, Instituto tecnológico y de Estudios Superiores de Monterrey. (s.f.). <http://www.sistema.itesm.mx/va/dide/inf-doc/esraegias/>.
- Estayno, M., & Meles, J. (2014). El Rol del Product Owner en la Definición y Validación de las User Stories. *Revista Ciencia y Tecnología de la Facultad de Ingeniería - Universidad de Palermo*, 14.
- Evans, E. (2004). *Domain-Driven Design tackling Complexity in the Heart of Software*. Estados Unidos: Addison-Wesley.
- Fowler, M. (13 de 12 de 2005). <http://martinfowler.com/articles/newMethodology.html>. Obtenido de <http://dcc.uab.es/sites/default/files/21290/teoria/Fow2005-TheNewMethodology.pdf>
- Freeman, E. (1984). *Strategic Management: A Stakeholders Approach*. Boston: Pitmaan Press.
- Freeman, P. (1987). *Software Perspectives: The System Is the Message*. Addison Wesley.
- Gatica-Lara, F., & Uribarren-Berrueta, T. (17 de 09 de 2012). *Pautas en educación médica ¿Cómo elaborar una rúbrica?* (Elsevier, Ed.) Recuperado el 07 de 11 de 2018, de Investigación en Educación Médica: <http://www.scielo.org.mx/pdf/iem/v2n5/v2n5a10.pdf>

- Gause, D. &. (1989). *Exploring requirements: Quality before Design*. New York, Estados Unidos: Dorset house.
- Henrik, K. (2007). *SCRUM y XP desde las Trincheras*. New York: Info Q Enterprise Software Development Series.
- Humphrey, W. (1995). *A discipline for Software Engineering*. Estados Unidos: Addison-wesley.
- IEEE Computer Society. (2004). *SWEBOK (R) Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society.
- Institute of Electrical and Electronial Engineer. (1983). *IEEE Standar Glossary of Software Terminology*. New York.
- Jacobson, I. (1994). *Object Oriented Software Engineering*. Estados Unidos: Addison Wesley.
- Jacobson, I. B. (2000). *El Proceso Unificado de Desarrollo*. Madrid: Addison Wesley.
- Kovitz, B. (1988). *Practical Software Requeriments- A Manial of Content and Style*. Estados Unidos: Manning.
- Kuechler, M. (1988). *The Survey Method: An Indispensable Tool for Social Sciences Everywhere?* Florida: American Behavioralist Scientist.
- Larman, C. (07 de 03 de 2017). *Larman's Laws of Organizational Behavior*. Obtenido de http://www.craiglarman.com/wiki/index.php?title=Larman%27s_Laws_of_Organizational_Behavior
- Lema, V. Z. (1986). *Conversaciones con Enrique Pichon-Rivière*. Buenos Aires: Ediciones Cinco S.A.
- Letelier, P., & Penadés, M. (15 de 01 de 2006). *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Obtenido de Ciencia y Técnica Administrativa: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- M., L. M. (2009). Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE*, 1060-1076.
- MAXWELL, J. A. (1996). *QUALITATIVE RESEARCH DESIGN. An Interactive Approach*. California: Sage Publicatios.
- McConnell, S. (1996). *Rapid Development*. Estados Unidos: Microsoft Press.
- McConnell, S. (Enero-Febrero de 2001). Who needs Software Engineering? *IEEE Software*, 5-8.
- Morales Bueno, P. y. (2004). Aprendizaje basado en problemas . *Theoria*, Vol. 13, 145-157.
- NONAKA, I. y. (1995). *The knowledge-creating company: How Japanese Companies Create the Dynamics of Innovation*. New York-Oxford: H.Oxford University Press.
- Pichon-Rivière, E. (2002). *Teoría del Vínculo* (1ra. reimpresión 24 ed.). Buenos Aires: Nueva Visión.
- Pressman, R. (2005). *Ingeniería del Software. Un enfoque práctico* (Sexta ed.). Mc Graw - Hill Interamericana.
- Rivière, E. P. (1997). *El Proceso Grupal: del Psicoanálisis a la Psicología Social*. Buenos Aires: Coleccion Psicología Contemporanea.

- Rubin, K. (2013). *Essential Scrum- A practical guide to the most popular agile process*. Michigan, Estados Unidos: Addison Wesley.
- Rumbaugh, J. (1996). *Modelado y Diseño Orientado a Objetos*. Prentice Hall.
- Rupp, C. (Mayo/Junio de 2002). Los requerimientos y la Psicología. *IEEE Software*, 22-24.
- SAINZ DE ABAJO, B., DE LA TORRE DíEZ, I., & LÓPEZ-CORONADO, M. (s.f.). APLICACIÓN DE LA METODOLOGÍA ABP. Ventajas del Aprendizaje Autodirigido.
- Sampieri, R. H. (2014). *Metodología de la Investigación* (6 edición ed.). Mexico D.F.: Mc Graw Hill Education.
- Schwaber, K. &. (2001). *Agile Software Development with Scrum*. London, Estados Unidos: Prentice Hall.
- Schwaber, K. &. (Julio de 2013). *The Scrum Primer - The Definitive Guide to Scrum: The Rules of the Game*. Recuperado el 12 de 07 de 2014, de <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf>
- Silberschatz, A., Gagne, G., & Galvin, P. B. (2006). *Fundamentos de Sistemas Operativos* (Vol. 1). Madrid, España: McGraw-Hill Interamericana.
- Software Engineering Institute. (10 de 2010). *Software Engineering Institute - Carnegui Mellon University*. Obtenido de www.sei.cmu.edu: www.sei.cmu.edu/reports/10tr033.pdf
- Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). Mexico: Addison- Wesley.
- V., L. P. (1995). *System Requirements Engineering*. McGraw-Hill International series in Software Engineering.
- Watts, H. (1995). *A Discipline for Software Engineering*. New York: Addison Wesley.
- www.trabajo.gob.ar. (Mayo de 2012). Obtenido de Ministerio de Trabajo de la Nación Argentina: http://www.trabajo.gob.ar/capacitacion/calidad/utecc_mapas.asp
- Yourdon, E. (1982). *Análisis Estructurado Moderno*. Prentice Hall Hispanoamericana S.A.

ANEXOS

ANEXO I. Modelo de Encuesta

1. Edad: 1. Menos de 26 2. Entre 26 y 32 3. Entre 33 y 39 4. Entre 40 y 45 5. Más de 46
2. Sexo: 1. Mujer 2. Hombre
3. Formación o Título: 1. Ingeniero en Sistemas de Información o similar 2. Licenciado en Informática o similar 3. Analista Universitario de Sistemas 4. Otro, indique cuál
4. Localidad donde trabaja: 1. Ciudad Autónoma de Buenos Aires 2. Pcia. de Buenos Aires 3. Córdoba 4. Rosario 5. Otra, cuál
5. Actividad Laboral:
6. Antigüedad en la última función de trabajo:
7. ¿El equipo de desarrollo en el que trabaja, está co-locado o distribuido? 1. Co-locado 2. Distribuido 3. Otra modalidad, ¿cuál?
8. ¿Cuáles son las tres tareas principales que usted desarrolla en su trabajo? 1. Gestión de Proyecto 2. Ingeniería de Requerimientos 3. Arquitectura 4. Diseño 5. Implementación 6. Testing 7. Despliegue 8. ¿Otras? ¿Cuáles?

<p>9. En el ámbito del equipo de desarrollo en el que trabaja, utiliza metodologías basadas en proceso definidos, si es así ¿cuáles?</p> <ol style="list-style-type: none"> 1. PUD (Proceso Unificado de Desarrollo) 2. RUP (Rational Unified Process) 3. ICONIX 4. Yourdon 5. Microsoft Solution Framework 6. No utilizo metodologías basadas en procesos definidos 7. ¿Otras? ¿Cuáles?
<p>10. En el ámbito del equipo de desarrollo en el que trabaja, utiliza metodologías/frameworks basados en filosofía Lean/Ágil, si es así ¿cuáles?</p> <ol style="list-style-type: none"> 1. Scrum 2. XP 3. FDD 4. Kanban 5. Scrumban 6. ATDD/TDD 7. No utilizo metodologías basadas en metodologías Lean/Ágiles 8. ¿Otras?
<p>11. ¿Qué metodología de gestión de proyectos utiliza el equipo de desarrollo en el que trabaja?</p> <ol style="list-style-type: none"> 1. Metodología de gestión basada en PMI 2. Metodología de gestión basada en CMMI 3. Metodología de gestión ágil 4. Metodología de gestión lean 5. Metodología de gestión basada en RUP 6. No utilizo metodologías de gestión de Proyectos 7. Otras, ¿cuáles?
<p>12. ¿En los equipos de desarrollo de software en los cuáles trabajó o trabaja actualmente, qué competencias técnicas son requeridas? <i>Puede marcar más de una.</i></p> <ol style="list-style-type: none"> 1. Programación o Desarrollo 2. Manejo de Técnicas de Ingeniería de Requerimientos (Casos de Uso, Técnicas de Elicitación) 3. Gestión ágil de requerimientos (user stories) 4. Testing de Software 5. Gestión de Personas 6. Gestión Ágil de Proyectos 7. Gestión Tradicional de Proyectos 8. Modelado de Arquitecturas de Software 9. Diseño de Interacción Humano-Máquina 10. Manejo de Herramientas para Testing Automatizado 11. Manejo de Herramientas para Gestión de Configuración/Integración Continua 12. Otras, ¿cuáles?
<p>13. ¿En los equipos de desarrollo de software en los cuáles trabajó o trabaja actualmente, qué competencias conductuales son requeridas? <i>Puede marcar más de una.</i></p> <ol style="list-style-type: none"> 1. Resolución de problemas complejos 2. Trabajo en Equipo 3. Autoorganización 4. Manejo de Inglés 5. Pensamiento Crítico 6. Inteligencia Emocional 7. Orientación a Servicios 8. Negociación 9. Flexibilidad Cognitiva 10. Otras, ¿cuáles?

<p>14. ¿Cuál es la formación más adecuada para los integrantes de un equipo de desarrollo de software?</p> <ol style="list-style-type: none"> 1. Título Universitario de Grado 2. Título Universitario Intermedio 3. Tecnicatura 4. Estudios no formales 5. Estudiantes de carreras universitarias 6. La formación no afecta en la conformación de los equipos de desarrollo
<p>15. Según su criterio ¿Cuáles son los aspectos que conspiran contra la obtención de un producto de software de calidad?</p> <ol style="list-style-type: none"> 1. Falta de gestión 2. Falta de comunicación 3. Falta de liderazgo 4. Falta de conocimiento técnico 5. Escaso o nulo nivel de involucramiento del Cliente 6. Malas relaciones interpersonales 7. Malas estimaciones 8. Compromisos asumidos con poca información 9. Subestimación del producto a construir 10. Mala o escasa adaptación de la metodología o proceso a utilizar 11. Otras, ¿cuáles?
<p>16. Según su criterio, ¿cuál de los siguientes aspectos de la relación vincular entre los involucrados en el desarrollo de un producto de software, es necesario mejorar?</p> <ol style="list-style-type: none"> 1. <i>Comunicación</i> 2. <i>Compromiso</i> 3. <i>Involucramiento</i> 4. <i>Cooperación</i> 5. <i>Otro, cuál</i>
<p>17. ¿Qué recomendaciones haría usted para la formación de un equipo de desarrollo de software eficiente?</p>

ANEXO II. Tablas de Frecuencia

1. Tabla de frecuencias Variables edad/ Antigüedad en la última función

Años/ Antigüedad	0	0,5	1	3	4	5	6	7	8	9	10	11	17	2	20
1. Menos de 26	1	0	4	0	1	0	0	0	0	0	0	0	0	2	0
2. Entre 26 y 32	0	0	6	2	1	2	0	2	0	0	1	1	0	4	0
3. Entre 33 y 39	0	0	1	1	0	0	1	2	0	1	0	1	0	5	0
4. Entre 40 y 45	0	0	3	2	0	1	0	2	0	1	2	0	0	1	1
5. Más de 46	0	1	0	0	2	0	0	0	1	0	0	0	1	0	0

2. Tabla de Frecuencias: edad y localización del Equipo de desarrollo en el que se desempeña

	. Co-locado	Distribuido	. Otra modalidad, ¿Cuál?	Desconozco	Mixto	Principalment e co-locado con miembros distribuidos en proyectos particulares	algunos días 1. y otros 2.
Menos de 26	5	2	0	1	0	0	0
Entre 26 y 32	10	7	1	0	0	1	0
Entre 33 y 39	3	7	0	0	1	0	1
Entre 40 y 45	8	5	0	0	0	0	0
Más de 46	3	2	0	0	0	0	0

3. Tabla de contingencia (Antigüedad en la última función de trabajo y Localización del Equipo de Desarrollo en el que se desempeña)

Antigüedad/ Localización del Equipo	Co-Locado	Distribuido	Otra modalidad	Desconozco	Mixto	Principalmente co-locado pero con algunos	Algunos días colocados y otros distribuido
0	1	0	0	0	0	0	0
0.5	1	0	0	0	0	0	0
1	8	4	1	0	1	0	0
10	2	0	0	0	0	1	0
11	2	0	0	0	0	0	0
17	1	0	0	0	0	0	0
2	5	6	0	0	0	0	1
20	0	1	0	0	0	0	0
3	1	4	0	0	0	0	0
4	2	1	0	1	0	0	0
5	2	1	0	0	0	0	0
6	0	1	0	0	0	0	0
7	3	3	0	0	0	0	0
8	0	1	0	0	0	0	0
9	1	1	0	0	0	0	0

4. Tabla de Frecuencia para las variables tres principales que realiza en el trabajo? y Localización del equipo de desarrollo en el que trabaja

	Co-Locado	Distribuido	Otra Modalidad	Desconozco	Mixto	Principal- mente co- locado	Algunos días Co- locado Otros Distribuido
1. Gestión de Proyecto	2	1	0	0	0	0	0

1. Gest Proyecto, Ing Requer Arquitectura	0	2	0	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 4. Diseño	0	1	0	0	0	0	0
1. Gest Proy, 2. Ing . Requer 5. Implementación	1	1	0	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, Pruebas	0	1	0	0	0	0	0
1. Gestión de Proyecto, 3. Arquitectura, 4. Diseño	1	0	0	0	0	0	0
1. Gestión de Proyecto, 3. Arquitectura, 5. Implementación	1	2	0	0	0	0	0
1. Gestión de Proyecto, 4. Diseño, 5. Implementación	0	3	0	0	0	0	0
1. Gestión de Proyecto, 5. Implementación	1	0	0	0	0	0	0
1. Gestión de Proyecto, 5. Implementación, 6. Despliegue	1	0	1	0	0	0	0
1. Gestión de Proyecto, 6. Despliegue, People management	0	1	0	0	0	0	0
1. Gest Proyecto, Consultoría sobre adecuaciones de obs de TI	1	0	0	0	0	0	0
1. Gestión de Proyecto, Gestión de personas, Gestión de ingeniería	0	1	0	0	0	0	0
2. Ingeniería de Requerimientos	0	0	0	1	0	0	0
2. Ingeniería de Requerimientos, 3. Arquitectura, 4. Diseño	0	1	0	0	0	0	0
2. Ingeniería de Requerimientos, 4. Diseño, 5. Implementación	1	0	0	0	0	0	0
2. Ingeniería de Requerimientos, 4. Diseño, 6. Despliegue	1	0	0	0	0	0	0
2. Ingeniería de Requerimientos, 5. Implementación, 6. Despliegue	0	0	0	0	0	0	1
2. Ingeniería de Requerimientos, 5. Implementación, Qual Assur	0	1	0	0	0	0	0
3. Arquitectura, 4. Diseño, 5. Implementación	7	5	0	0	0	0	0
4. Diseño, 5. Implementación	1	1	0	0	0	0	0
4. Diseño, 5. Implementación, 6. Despliegue	1	0	0	0	0	0	0
4. Diseño, 5. Implementación, 7. Otras, ¿Cuáles?	1	0	0	0	0	0	0
5. Implementación	1	0	0	0	0	0	0
5. Implementación, 6. Despliegue	1	0	0	0	0	0	0
5. Implementación, 6. Despliegue, Consultoría	1	0	0	0	0	0	0
6. Despliegue, Mejora de procesos	1	0	0	0	0	0	0
7. Otras, ¿Cuáles?, Gestión de Auditores Externos Interno	0	1	0	0	0	0	0
Gestión de Proyecto, Arquitectura, Despliegue	0	0	0	0	0	1	0
Gestión de Proyecto, Arquitectura, Implementación	0	1	0	0	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Despliegue	0	0	0	0	1	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Diseño	1	0	0	0	0	0	0
Gestión de Proyecto, Ing de Requerimientos, Gestión de recursos	1	0	0	0	0	0	0
Gest Proy, Ing Requ Gest equipo a cargo de Ges Pro e Ing Req	1	0	0	0	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Ing Procesos	1	0	0	0	0	0	0
Testing	1	0	0	0	0	0	0

5. Tabla de frecuencias para las variables uso de metodologías/frameworks basados en filosofía Lean Ágil y localización del equipo de desarrollo en el que trabaja

Metodología/localización	Co Locado	Distribuido	Otra modalidad	Desconozco	Mixto	Principalment e Locado	Algunos días locado otros distribuido
--------------------------	-----------	-------------	----------------	------------	-------	------------------------	---------------------------------------

1. Scrum	11	13	0	1	1	1	0
1. Scrum, 2. XP, 3. Kanban	0	1	0	0	0	0	0
1. Scrum, 2. XP, 4. ATDD/TDD	0	2	0	0	0	0	0
1. Scrum, 3. Kanban	7	0	0	0	0	0	0
1. Scrum, 3. Kanban, 4. ATDD/TDD	0	1	0	0	0	0	0
1. Scrum, 4. ATDD/TDD	0	1	0	0	0	0	0
1. Scrum, 4. ATDD/TDD,	0	1	0	0	0	0	0
3. Kanban	1	0	1	0	0	0	0
3. Kanban, 4. ATDD/TDD	0	0	0	0	0	0	1
5. No utilizo metodologías basadas en Lean/Ágiles	8	3	0	0	0	0	0
6. Otras, ¿Cuáles?	0	1	0	0	0	0	0

6. Tabla de Frecuencias para las variables metodología de Gestión de Proyectos y Localización del equipo de desarrollo en el que trabaja

	Co Locado	Distribuido	Otra Modalidad	Desconozco	Mixto	Principalmente co Locado	Algunos días co locado y otras distribuido
1. Metodología de gestión basada en PMI	6	4	0	0	0	0	0
2. Metodología de gestión ágil	14	15	1	1	1	0	0

3. Metodología de gestión lean	1	1	0	0	0	0	0
4. Metodología de gestión basada en RUP	3	1	0	0	0	0	0
5. No utilizo metodologías de gestión de proyectos	4	1	0	0	0	0	1
Ciclo de vida propio, combinación de "PMI" con ágil	0	1	0	0	0	0	0
Híbridos	1	0	0	0	0	0	0
Ninguna de las anteriores	0	0	0	0	0	1	0

7. Tablas de Frecuencia para las variables competencias técnicas requeridas y Localización del Equipo de desarrollo en el que trabaja,

	Co Locado	Distribuido	Otra modalidad	Desconozco	Mixto	Principalmente co locado	Algunos días co locado otros distribuido
1. Programación o Desarrollo	3	1	0	0	0	0	0
1. Progr o Des, 2. Gest ágil de requer 3. Testing de Soft	0	1	0	0	0	0	0
1. Prog o Des, 2. Gest ágil de reque 3. Testing de Soft 4. Gest Pers	0	2	0	0	0	0	0
1. ProgrDes 2. Gestágil requerimientos 3. Testing de Soft, 4. Gest Pers 5. GestÁgil Proy, 6. GestTrad Proy 7. Mod de ArqSoft 8. DisInterHum-Máqui, 9. Manejo Herram Gestde Config/Integ Continua, DevOps, oper, gestde la configuración	0	1	0	0	0	0	0
1. Progr o Des, 2. Gest ágil de requer 3. Testing de Soft, 4. Gestde Personas, 5. Gestión Ágil de Proyectos, 6. Gestión Tradicional de Proyectos, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	1	0	0	0	0	0

1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 4. Gestión de Personas, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software	0	1	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 4. Gestión de Personas, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software, 8. Diseño de Interacción Humano-Máquina, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	2	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 4. Gestión de Personas, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	2	1	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 4. Gestión de Personas, 5. Gestión Ágil de Proyectos, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	1	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software	1	0	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software, 8. Diseño de Interacción Humano-Máquina, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	1	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	0	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 5. Gestión Ágil de Proyectos, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	2	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 7. Modelado de Arquitecturas de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	1	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 8. Diseño de Interacción Humano-Máquina	0	0	0	0	0	0	1

1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 3. Testing de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	2	0	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 4. Gestión de Personas, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	1	0	0	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 5. Gestión Ágil de Proyectos	0	0	0	1	0	0	0
1. Programación o Desarrollo, 2. Gestión ágil de requerimientos (user stories), 5. Gestión Ágil de Proyectos, 8. Diseño de Interacción Humano-Máquina, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	0	0	0	0	0	0
1. Programación o Desarrollo, 3. Testing de Software	1	0	1	0	0	0	0
1. Programación o Desarrollo, 3. Testing de Software, 4. Gestión de Personas, 5. Gestión Ágil de Proyectos, 8. Diseño de Interacción Humano-Máquina	0	1	0	0	0	0	0
1. Programación o Desarrollo, 3. Testing de Software, 4. Gestión de Personas, 6. Gestión Tradicional de Proyectos, 7. Modelado de Arquitecturas de Software	1	0	0	0	0	0	0
1. Programación o Desarrollo, 3. Testing de Software, 4. Gestión de Personas, 7. Modelado de Arquitecturas de Software, 8. Diseño de Interacción Humano-Máquina	1	0	0	0	0	0	0
1. Programación o Desarrollo, 3. Testing de Software, 5. Gestión Ágil de Proyectos, 6. Gestión Tradicional de Proyectos	1	0	0	0	0	0	0
1. Programación o Desarrollo, 3. Testing de Software, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software	1	0	0	0	0	0	0
1. Programación o Desarrollo, 3. Testing de Software, 7. Modelado de Arquitecturas de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	1	0	0	0	0	0
1. Programación o Desarrollo, 4. Gestión de Personas, 5. Gestión Ágil de Proyectos, 7. Modelado de Arquitecturas de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	1	0	0	0	0	0
1. Programación o Desarrollo, 4. Gestión de Personas, 6. Gestión Tradicional de Proyectos, 7. Modelado de Arquitecturas de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	0	0	0	0	0	0

1. Programación o Desarrollo, 4. Gestión de Personas, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	0	0	0	0	0	0
1. Programación o Desarrollo, 5. Gestión Ágil de Proyectos	1	0	0	0	0	0	0
1. Programación o Desarrollo, 7. Modelado de Arquitecturas de Software	1	0	0	0	0	0	0
1. Programación o Desarrollo, 7. Modelado de Arquitecturas de Software, 8. Diseño de Interacción Humano-Máquina, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	0	0	0	0	0	0
1. Programación o Desarrollo, 7. Modelado de Arquitecturas de Software, 9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	1	0	0	0	0	0
2. Gestión ágil de requerimientos (user stories), 5. Gestión Ágil de Proyectos, 6. Gestión Tradicional de Proyectos, 8. Diseño de Interacción Humano-Máquina	1	0	0	0	0	0	0
4. Gestión de Personas, 6. Gestión Tradicional de Proyectos, 7. Modelado de Arquitecturas de Software	0	1	0	0	0	0	0
6. Gestión Tradicional de Proyectos	1	0	0	0	0	0	0
9. Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	1	0	0	0	0	0
Manejo de Técnicas de Ingeniería de Requerimientos (Casos de Uso, Técnicas de Elicitación), Gestión ágil de requerimientos (user stories), Testing de Software, Gestión de Personas, Gestión Tradicional de Proyectos	1	0	0	0	0	0	0
Programación o Desarrollo, Gestión ágil de requerimientos (user stories), Manejo de Herramientas para Gestión de Configuración/Integración Continua	0	0	0	0	0	1	0
Programación o Desarrollo, Gestión ágil de requerimientos (user stories), Testing de Software, Gestión de Personas, Gestión Ágil de Proyectos, Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	0	0	0	0	0	0
Programación o Desarrollo, Gestión ágil de requerimientos (user stories), Testing de Software, Gestión de Personas, Gestión Ágil de Proyectos, Modelado de Arquitecturas de Software, Diseño de Interacción Humano-Máquina, Manejo de Herramientas para Testing Automatizado, Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	0	0	0	0	0	0
Programación o Desarrollo, Gestión ágil de requerimientos (user stories), Testing de Software, Gestión Ágil de Proyectos, Gestión Tradicional de Proyectos, Diseño de Interacción Humano-Máquina,	0	0	0	0	1	0	0

Manejo de Herramientas para Gestión de Configuración/Integración Continua							
Programación o Desarrollo, Gestión ágil de requerimientos (user stories), Testing de Software, Manejo de Herramientas para Testing Automatizado	0	1	0	0	0	0	0
Programación o Desarrollo, Manejo de Técnicas de Ingeniería de Requerimientos (Casos de Uso, Técnicas de Elicitación), Gestión ágil de requerimientos (user stories), Testing de Software, Gestión de Personas, Gestión Ágil de Proyectos, Gestión Tradicional de Proyectos, Modelado de Arquitecturas de Software, Diseño de Interacción Humano-Máquina, Manejo de Herramientas para Testing Automatizado, Manejo de Herramientas para Gestión de Configuración/Integración Continua	1	0	0	0	0	0	0

8.- Tabla de frecuencias para las variables principales que realiza y Sexo

	1.	
	Mujer	2. Hombre
1. Gestión de Proyecto	0	3
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 3. Arquitectura	1	1
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 4. Diseño	0	1
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 5. Implementación	1	1
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, Pruebas	1	0
1. Gestión de Proyecto, 3. Arquitectura, 4. Diseño	0	1
1. Gestión de Proyecto, 3. Arquitectura, 5. Implementación	0	3
1. Gestión de Proyecto, 4. Diseño, 5. Implementación	1	2
1. Gestión de Proyecto, 5. Implementación	1	0
1. Gestión de Proyecto, 5. Implementación, 6. Despliegue	0	2
1. Gestión de Proyecto, 6. Despliegue, People management	1	0
1. Gestión de Proyecto, 7. Otras, ¿Cuáles?, Consultoría sobre adecuaciones de obs de TI	0	1
1. Gestión de Proyecto, Gestión de personas, Gestión de ingeniería	0	1
2. Ingeniería de Requerimientos	1	0
2. Ingeniería de Requerimientos, 3. Arquitectura, 4. Diseño	0	1
2. Ingeniería de Requerimientos, 4. Diseño, 5. Implementación	0	1
2. Ingeniería de Requerimientos, 4. Diseño, 6. Despliegue	1	0

2. Ingeniería de Requerimientos, 5. Implementación, 6. Despliegue	0	1
2. Ingeniería de Requerimientos, 5. Implementación, Quality Assurance	0	1
3. Arquitectura, 4. Diseño, 5. Implementación	0	12
4. Diseño, 5. Implementación	0	2
4. Diseño, 5. Implementación, 6. Despliegue	0	1
4. Diseño, 5. Implementación, 7. Otras, ¿Cuáles?	0	1
5. Implementación	1	0
5. Implementación, 6. Despliegue	1	0
5. Implementación, 6. Despliegue, Consultoría	0	1
6. Despliegue, Mejora de procesos	1	0
7. Otras, ¿Cuáles?, Gestión de Auditores Externos Interno	1	0
Gestión de Proyecto, Arquitectura, Despliegue	0	1
Gestión de Proyecto, Arquitectura, Implementación	0	1
Gestión de Proyecto, Ingeniería de Requerimientos, Despliegue	1	0
Gestión de Proyecto, Ingeniería de Requerimientos, Diseño	1	0
Gestión de Proyecto, Ingeniería de Requerimientos, Gestión de recursos	0	1
Gestión de Proyecto, Ingeniería de Requerimientos, Gestión de equipo que tiene a su cargo las actividades de Gestión de Proyecto y de Ingeniería de Requerimientos	1	0
Gestión de Proyecto, Ingeniería de Requerimientos, Ingeniería de Procesos	1	0
Testing	1	0

9.- Tabla de Frecuencias para las variables tareas principales que realiza en su trabajo y si utiliza metodologías basadas en procesos definidos

	1. PUD (Proceso Unificado de Desarroll o)	1. PUD (Proceso Unificado de Desarrollo), 2. RUP (Rational Unified Process)	2. RUP (Ration al Unified Proces s)	3. No utilizo metodología s basadas en procesos definidos	4. Otras , indiqu e ¿cuál es?	4. Otras, ¿cuáles?, Magerit, COBIT, ITIL, ISO	4. Otras, ¿cuále s?, Scrum	A g il s	No se utilizan metodología s basadas en procesos definidos.	PUD (Proceso Unificado de Desarrollo), RUP (Rational Unified Process)	Una metodología ágil definida por el equipo de desarrollo	parte s de meto dolog ía ágil
1. Gestión de Proyecto	1	0	0	2	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 3. Arquitectura	0	1	0	1	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 4. Diseño	0	0	0	1	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 5. Implementación	0	1	0	1	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, Pruebas	0	0	0	1	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 3. Arquitectura, 4. Diseño	1	0	0	0	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 3. Arquitectura, 5. Implementación	0	0	0	3	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 4. Diseño, 5. Implementación	0	0	0	1	0	0	1	1	0	0	0	0
1. Gestión de Proyecto, 5. Implementación	0	0	1	0	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 5. Implementación, 6. Despliegue	0	0	0	2	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 6. Despliegue, People management	0	0	0	1	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 7. Otras, ¿Cuáles?, Consultoría sobre adecuaciones de obs de TI	0	0	0	0	0	1	0	0	0	0	0	0
1. Gestión de Proyecto, Gestión de personas, Gestión de ingeniería	0	0	0	1	0	0	0	0	0	0	0	0
2. Ingeniería de Requerimientos	0	0	0	1	0	0	0	0	0	0	0	0
2. Ingeniería de Requerimientos, 3. Arquitectura, 4. Diseño	0	0	0	1	0	0	0	0	0	0	0	0
2. Ingeniería de Requerimientos, 4. Diseño, 5. Implementación	0	0	0	1	0	0	0	0	0	0	0	0
2. Ingeniería de Requerimientos, 4. Diseño, 6. Despliegue	1	0	0	0	0	0	0	0	0	0	0	0
2. Ingeniería de Requerimientos, 5. Implementación, 6. Despliegue	0	0	0	0	0	0	0	0	0	0	0	1

3. Arquitectura, 4. Diseño, 5. Implementación	1	0	1	8	1	0	0	0	0	0	0	1	0
4. Diseño, 5. Implementación	0	0	0	2	0	0	0	0	0	0	0	0	0
4. Diseño, 5. Implementación, 6. Despliegue	0	0	0	1	0	0	0	0	0	0	0	0	0
4. Diseño, 5. Implementación, 7. Otras, ¿Cuáles?	0	0	0	1	0	0	0	0	0	0	0	0	0
5. Implementación	0	0	0	1	0	0	0	0	0	0	0	0	0
5. Implementación, 6. Despliegue	0	0	0	0	1	0	0	0	0	0	0	0	0
5. Implementación, 6. Despliegue, Consultoría	0	0	0	1	0	0	0	0	0	0	0	0	0
6. Despliegue, Mejora de procesos	0	0	0	1	0	0	0	0	0	0	0	0	0
7. Otras, ¿Cuáles?, Gestión de Auditores Externos Interno	0	0	0	1	0	0	0	0	0	0	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Diseño	0	0	0	0	0	0	0	0	0	1	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Gestión de equipo que tiene a su cargo las actividades de Gestión de Proyecto y de Ingeniería de Requerimientos	0	0	0	0	0	0	0	0	0	0	1	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Ingeniería de Procesos	0	0	0	0	0	0	0	0	0	0	1	0	0
Testing	1	0	0	0	0	0	0	0	0	0	0	0	0

10. Tabla de Frecuencias para las variables tareas principales que realiza competencias conductuales son requeridas

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad	ae	af	ag	ah	ai	aj	ak	al			
1. Gestión de Proyecto	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 3. Arquitectura	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 4. Diseño	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	Comunicación	Compromiso	Cooperación	Involucramiento	Otro
1. Gestión de Proyecto	3	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 3. Arquitectura	1	0	0	1	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 4. Diseño	1	0	0	0	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, 5. Implementación	0	0	1	1	0
1. Gestión de Proyecto, 2. Ingeniería de Requerimientos, Pruebas	0	1	0	0	0
1. Gestión de Proyecto, 3. Arquitectura, 4. Diseño	1	0	0	0	0
1. Gestión de Proyecto, 3. Arquitectura, 5. Implementación	2	0	1	0	0
1. Gestión de Proyecto, 4. Diseño, 5. Implementación	2	0	1	0	0
1. Gestión de Proyecto, 5. Implementación	1	0	0	0	0
1. Gestión de Proyecto, 5. Implementación, 6. Despliegue	2	0	0	0	0
1. Gestión de Proyecto, 6. Despliegue, People management	1	0	0	0	0
1. Gestión de Proyecto, 7. Otras, ¿Cuáles?, Consultoría sobre adecuaciones de obs de TI	1	0	0	0	0
1. Gestión de Proyecto, Gestión de personas, Gestión de ingeniería	0	1	0	0	0
2. Ingeniería de Requerimientos	1	0	0	0	0
2. Ingeniería de Requerimientos, 3. Arquitectura, 4. Diseño	0	0	0	1	0
2. Ingeniería de Requerimientos, 4. Diseño, 5. Implementación	1	0	0	0	0
2. Ingeniería de Requerimientos, 4. Diseño, 6. Despliegue	1	0	0	0	0
2. Ingeniería de Requerimientos, 5. Implementación, 6. Despliegue	1	0	0	0	0
2. Ingeniería de Requerimientos, 5. Implementación, Quality Assurance	1	0	0	0	0
3. Arquitectura, 4. Diseño, 5. Implementación	9	1	0	0	2
4. Diseño, 5. Implementación	0	2	0	0	0
4. Diseño, 5. Implementación, 6. Despliegue	1	0	0	0	0
4. Diseño, 5. Implementación, 7. Otras, ¿Cuáles?	1	0	0	0	0
5. Implementación	0	1	0	0	0
5. Implementación, 6. Despliegue	1	0	0	0	0
5. Implementación, 6. Despliegue, Consultoría	1	0	0	0	0
6. Despliegue, Mejora de procesos	1	0	0	0	0
7. Otras, ¿Cuáles?, Gestión de Auditores Externos Interno	0	0	1	0	0

Gestión de Proyecto, Arquitectura, Despliegue	1	0	0	0	0
Gestión de Proyecto, Arquitectura, Implementación	1	0	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Despliegue	0	1	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Diseño	1	0	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Gestión de recursos	0	1	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Gestión de equipo que tiene a su cargo las actividades de Gestión de Proyecto y de Ingeniería de Requerimientos	1	0	0	0	0
Gestión de Proyecto, Ingeniería de Requerimientos, Ingeniería de Procesos	1	0	0	0	0
Testing	1	0	0	0	0

12.- Tabla de Frecuencias para las variables competencias conductuales y competencias técnicas requeridas

	G A P	G A R	G T P	M A S	M T H	P O I	P D S
1. Resolución de problemas complejos, 2. Trabajo en Equipo	0	0	0	0	0	0	1
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización	0	0	0	0	0	0	1
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico	0	1	0	0	0	0	2
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 6. Orientación a Servicios	1	0	0	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 6. Orientación a Servicios, 7. Negociación	0	5	0	0	0	0	3
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 7. Negociación	0	2	0	0	0	0	1
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 7. Negociación, Comunicación, liderazgo, aprendizaje, flexibilidad, adaptación al cambio, entender el negocio	0	0	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 6. Orientación a Servicios	0	0	0	0	1	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 7. Negociación	0	0	0	0	0	0	2
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	0	0	0	1	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios, 7. Negociación	0	0	0	0	0	0	1
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 7. Negociación, Liderazgo	0	0	0	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 6. Orientación a Servicios, 7. Negociación	0	0	0	0	0	0	1
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 7. Negociación, Flexibilidad Cognitiva	0	0	0	0	0	0	1

1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, Inteligencia Emocional, 6. Orientación a Servicios, 7. Negociación, Flexibilidad Cognitiva, Personalmente, quisiera marcar la importancia de la Inteligencia Emocional, muy necesaria para lograr un buen trabajo en equipo, entender los sentimientos de los miembros del equipo y así saber cómo afrontar la comunicación correcta entre ellos. En los ambientes de trabajo de desarrollo de software la COMUNICACIÓN dentro y fuera del equipo es fundamental y lo que más dificultad presenta. Creo, que esto se ve influenciado directamente con la falta de inteligencia emocional de las personas, que en vez de afrontar con el dialogo aquellas situaciones o problemas que le hayan causado frustración o enojo, callan y generan brechas de comunicación que afecta directamente el entorno de trabajo y el trabajo en equipo.

1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 6. Orientación a Servicios	0	0	0	0	0	0	0	0	1	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	1	0	0	0	0	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 6. Orientación a Servicios, 7. Negociación	0	0	0	0	0	0	0	1	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 4. Manejo de Inglés, 5. Pensamiento Crítico	0	1	0	0	0	0	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 4. Manejo de Inglés, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	0	0	0	0	0	1	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 4. Manejo de Inglés, 6. Orientación a Servicios, 7. Negociación	0	0	1	0	0	0	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	0	0	0	0	0	0	0	0	1
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 6. Orientación a Servicios	0	0	1	0	0	0	0	0	0	0
1. Resolución de problemas complejos, 4. Manejo de Inglés, 6. Orientación a Servicios	0	0	0	0	0	0	1	0	0	0
2. Trabajo en Equipo, 3. Autoorganización	0	0	0	0	0	0	0	0	0	1
2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés	0	0	0	0	0	0	0	0	0	1
2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico	0	0	0	0	0	1	0	0	0	1
2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 6. Orientación a Servicios	0	1	0	0	0	0	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 7. Negociación	0	0	0	0	0	0	0	0	0	1
2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico	0	0	0	1	0	0	0	0	0	1
2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	0	0	0	0	0	0	0	0	1
2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios, 7. Negociación	0	1	0	0	0	0	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 7. Negociación	0	0	0	0	0	0	0	0	0	1
2. Trabajo en Equipo, 3. Autoorganización, 6. Orientación a Servicios	0	0	0	0	0	0	0	0	0	1
2. Trabajo en Equipo, 3. Autoorganización, Flexibilidad Cognitiva	0	0	0	0	0	0	0	0	0	1
2. Trabajo en Equipo, 5. Pensamiento Crítico	0	0	0	0	0	1	0	0	0	0
2. Trabajo en Equipo, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	1	0	0	0	0	0	0	0	0
2. Trabajo en Equipo, 6. Orientación a Servicios, 7. Negociación	0	0	1	0	0	0	0	0	0	0

GAP	GAR	GP	GTP	MAS	MH	MTI	POD	T S
-----	-----	----	-----	-----	----	-----	-----	--------

Gestión ágil de proyectos	Gestión ágil de requerimientos	Gestión personas	Gestión Tradicional de Proyectos	Modelado de Arquitecturas de Software	Manejo de Herramientas para Gestión de Configuración/Integración Continua	Manejo de Técnicas de Ingeniería de Requerimientos	Programación o Desarrollo
---------------------------	--------------------------------	------------------	----------------------------------	---------------------------------------	---	--	---------------------------

13.- Tabla de Frecuencias para las variables competencias conductuales requeridas y aspectos de la relación vincular a mejorar entre los involucrados en el desarrollo de un producto de software

	Comunicación	Compromiso	Cooperación	Involucramiento	Otro
1. Resolución de problemas complejos, 2. Trabajo en Equipo	0	1	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico	4	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 6. Orientación a Servicios	2	0	0	1	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 6. Orientación a Servicios, 7. Negociación	4	0	1	1	2
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 7. Negociación	4	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 7. Negociación, Comunicación, liderazgo, aprendizaje, flexibilidad, adaptación al cambio, entender el negocio	0	1	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 6. Orientación a Servicios	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 7. Negociación	1	1	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios, 7. Negociación	0	0	1	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 7. Negociación, Liderazgo	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 6. Orientación a Servicios, 7. Negociación	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, 7. Negociación, Flexibilidad Cognitiva	0	1	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico, Inteligencia Emocional, 6. Orientación a Servicios, 7. Negociación, Flexibilidad Cognitiva, Personalmente, quisiera marcar la importancia de la Inteligencia Emocional, muy necesaria para lograr un buen trabajo en equipo, entender los sentimientos de los miembros del equipo y así saber cómo afrontar la comunicación correcta entre ellos. En los ambientes de trabajo de desarrollo de software la COMUNICACIÓN dentro y fuera del equipo es fundamental y lo que más dificultad presenta. Creo, que esto se ve influenciado directamente con la falta	1	0	0	0	0

de inteligencia emocional de las personas, que en vez de afrontar con el dialogo aquellas situaciones o problemas que le hayan causado frustración o enojo, callan y generan brechas de comunicación que afecta directamente el entorno de trabajo y el trabajo en equipo.

1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 6. Orientación a Servicios	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	1	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 3. Autoorganización, 6. Orientación a Servicios, 7. Negociación	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 4. Manejo de Inglés, 5. Pensamiento Crítico	0	1	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 4. Manejo de Inglés, 5. Pensamiento Crítico, 6. Orientación a Servicios	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 4. Manejo de Inglés, 6. Orientación a Servicios, 7. Negociación	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 5. Pensamiento Crítico, 6. Orientación a Servicios	1	0	0	0	0
1. Resolución de problemas complejos, 2. Trabajo en Equipo, 6. Orientación a Servicios	0	0	1	0	0
1. Resolución de problemas complejos, 4. Manejo de Inglés, 6. Orientación a Servicios	0	0	0	1	0
2. Trabajo en Equipo, 3. Autoorganización	1	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés	0	0	1	0	0
2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 5. Pensamiento Crítico	3	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 6. Orientación a Servicios	1	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 4. Manejo de Inglés, 7. Negociación	1	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico	2	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	1	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 6. Orientación a Servicios, 7. Negociación	1	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 5. Pensamiento Crítico, 7. Negociación	1	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, 6. Orientación a Servicios	1	0	0	0	0
2. Trabajo en Equipo, 3. Autoorganización, Flexibilidad Cognitiva	1	0	0	0	0
2. Trabajo en Equipo, 5. Pensamiento Crítico	1	0	0	0	0
2. Trabajo en Equipo, 5. Pensamiento Crítico, 6. Orientación a Servicios	0	1	0	0	0
2. Trabajo en Equipo, 6. Orientación a Servicios, 7. Negociación	1	0	0	0	0