

# Un estudio sobre las prácticas de la ingeniería del *software* en Costa Rica: Resultados preliminares

## Reporte Técnico

Christian Quesada-López<sup>1, 2</sup>, Marcelo Jenkins<sup>1, 2</sup>

CITIC, Universidad de Costa Rica, San Pedro, Costa Rica<sup>1</sup>  
Ministerio de Ciencia, Tecnología y Telecomunicaciones, San José, Costa Rica<sup>2</sup>  
{cristian.quesadalopez, marcelo.jenkins}@ucr.ac.cr

**Resumen.** La caracterización de las prácticas de la ingeniería del *software* (SE) utilizadas por la industria permite entender la naturaleza y la madurez de las organizaciones de desarrollo y dirigir esfuerzos para el mejoramiento de su competitividad. El objetivo del presente estudio es caracterizar las prácticas de la ingeniería del *software* en la industria costarricense. Para alcanzar el objetivo se diseñó y ejecutó una encuesta en línea con 46 preguntas basadas en el *Software Engineering Body of Knowledge (SWEBOK)* y estudios previos en el área. Doscientos setenta y ocho profesionales participaron en la encuesta. Los resultados confirman que (1) la calidad y la productividad siguen siendo los principales desafíos. (2) Los profesionales requieren evidencia sobre las prácticas esenciales de la SE que generan mayor valor agregado a sus organizaciones. (3) Los principales desafíos los enfrentan en las actividades de aseguramiento de calidad y pruebas, y en la comunicación con la alta gerencia y los usuarios finales. (4) Las metodologías ágiles son las más utilizadas (46.7%), principalmente *Scrum* (24.5%), seguido de las metodologías tradicionales como cascada (19.9%), prototipos (19.0%), incremental (10.2%) y espiral (3.7%). (5) En el 85% de los casos el uso de estas dos metodologías es excluyente. (6) Los lenguajes más utilizados son los de la familia .Net (48%), seguido por Java (26%). (7) Las pruebas más comunes son las de unidad, de aceptación, funcionales, de sistema y de integración. (8) Las pruebas manuales prevalecen sobre las automatizadas, solo el 33% reportan el uso frecuente de pruebas automatizadas. La industria costarricense se encuentra en el momento justo para adoptar prácticas de aseguramiento de la calidad y mejoramiento y medición de procesos. Los resultados de la encuesta pueden ser de interés para la industria y la academia. Los resultados obtenidos confirman algunos de los reportados para otros países en estudios similares.

**Palabras clave.** Prácticas en la industria, Costa Rica, encuesta, Ingeniería de *software* experimental.

## 1 Introducción

El área de la ingeniería de software (SE) ha sido un campo de investigación activo desde 1968 y una disciplina que ha ido incrementando su nivel de madurez con el paso de los años. La IEEE estableció su comité para el desarrollo de estándares en la SE en 1976 y proporciona el *Software Engineering Body of Knowledge* (SWEBOK) como la guía base de conocimiento y prácticas relacionadas con la SE [SWE04]. Esta promueve el avance de las teorías de la SE y su aplicación práctica. El SWEBOK se compone de 15 áreas de conocimiento que describen el cómo desarrollar, administrar y dar mantenimiento al *software*, entre otros [BOU14].

La SE, por su naturaleza, requiere de la colaboración entre la industria y la academia para el desarrollo, mejoramiento y adopción de sus prácticas. Sin embargo, las colaboraciones industria y academia han sido muy limitadas en muchos países [GAR15]. Para disminuir las brechas existentes, la academia requiere conocer las prácticas actuales que le permitan mejorar los procesos de formación de los futuros profesionales y la industria requiere el apoyo de la academia para transferir a la práctica todas las nuevas tendencias tecnológicas emergentes [GAR15].

Múltiples investigadores han reportado estudios relacionadas con las prácticas de la SE en la industria, con el objetivo de incentivar las colaboraciones entre la industria y la academia y para entender las prácticas globales de la industria. La principal limitante de estos estudios es que se enfocan en distintas áreas de conocimiento y no pueden compararse entre sí. En la actualidad son necesarios estudios homogéneos sobre las prácticas de la SE en la industria que permitan la comparación de las prácticas comunes en diferentes países y regiones [GAR15].

En Costa Rica la industria del *software* se ha consolidado en los últimos 15 años y actualmente cuenta con más de 200 organizaciones de desarrollo de *software* [CAM14]. Los servicios de *software* representan el 1.31% del producto interno bruto (PIB) y las exportaciones del sector representan el segundo lugar en producción de divisas. En la actualidad, el país constituye el primer exportador per cápita de alta tecnología en la región y el cuarto en el mundo [SIL16]. Durante los últimos años, en el país se han realizado estudios para caracterizar y entender la evolución y el avance de la industria del *software* [MAT01] [BRE08] [CAM14] [PRO14] [BON15], sin embargo en nuestro conocimiento, ninguno enfocado específicamente en las prácticas de la SE.

La caracterización de las prácticas de la SE utilizadas por la industria permite entender la naturaleza y la madurez de las organizaciones de desarrollo de *software* y así dirigir esfuerzos para el mejoramiento de la competitividad del sector. Las organizaciones de desarrollo de *software* enfrentan grandes desafíos para mejorar la productividad y mejorar la calidad de los productos de *software*, y consideramos que las colaboraciones entre la industria y la academia son esenciales para mejorar este sector.

El objetivo de este estudio es caracterizar las prácticas de la SE en la industria de *software* costarricense, identificando las prácticas, métodos y herramientas más utilizadas en la práctica profesional y analizando los desafíos actuales que permitan potenciar las colaboraciones entre la industria y la academia. Además, este estudio compara algunas de las tendencias reportadas en estudios previos [GAR15] con los resultados obtenidos.

La estructura del artículo es la siguiente: la Sección 2 describe brevemente las prácticas de la SE consideradas en este estudio. Los estudios relacionados se resumen en la Sección 3. La Sección 4 describe el diseño del estudio. La Sección 5 analiza los resultados obtenidos y en la Sección 6 se discuten los hallazgos y las lecciones aprendidas. Finalmente, en la Sección 7 se presentan las conclusiones y el trabajo futuro.

## 2 Prácticas de la ingeniería del *software*

El *Software Engineering Body of Knowledge* (SWEBOK) establece una línea base de las prácticas de la ingeniería del *software* (SE) para impulsar el avance del conocimiento y la práctica en el área. El propósito del SWEBOK es proveer una caracterización de los límites de la disciplina que se encuentra en constante evolución [SWE04] [BOU14]. El SWEBOK se compone de 15 áreas de conocimiento que describen el cómo desarrollar, administrar y dar mantenimiento al *software*, entre otras, tal como se detalla en [BOU14] y se lista a continuación:

1. Requerimientos: fundamentos, proceso, elicitación, análisis, especificación, validación, consideraciones prácticas y herramientas.
2. Diseño: fundamentos, aspectos clave, estructura y arquitectura, interfaces de usuario, análisis y evaluación de la calidad, notaciones, estrategias y métodos y herramientas.
3. Construcción: fundamentos, administración, consideraciones prácticas, tecnologías y herramientas.
4. Pruebas: fundamentos, niveles, técnicas, mediciones, proceso y herramientas.
5. Mantenimiento: fundamentos, aspectos clave, proceso, técnicas y herramientas.
6. Administración de la configuración: proceso, identificación de la configuración, control, determinación del estado de la configuración, auditoría, administración de las entregas y liberaciones y herramientas.
7. Administración de la SE: alcance y definición, planificación, aprobación, revisión y evaluación, cierre, medición y herramientas.
8. Proceso de la SE: definición, ciclos de vida, evaluación y mejoramiento de procesos, medición y herramientas.
9. Modelos y métodos: modelaje, tipos, análisis y métodos de SE.
10. Calidad: fundamentos, proceso de aseguramiento de la calidad, consideraciones prácticas y herramientas.
11. Práctica profesional: carrera profesional, psicología y dinámica de grupos, habilidades de comunicación,
12. Economía de la SE: fundamentos, ciclos de vida, riegos, métodos de análisis y consideraciones prácticas.
13. Fundamentos computacionales: técnicas de resolución de problemas, abstracción, programación, lenguajes de programación, técnicas y herramientas de depuración, representación y estructuras de datos, algoritmos y complejidad, entre otros.
14. Fundamentos matemáticos: funciones, lógica, pruebas teóricas, conteo, grafos y árboles, probabilidad discreta, máquinas de estados, entre otros.
15. Fundamentos ingenieriles: métodos empíricos y de experimentación, análisis estadístico, medición, diseño, modelaje, simulación y prototipos, entre otros.

Dado el amplio rango de áreas de conocimiento, nuestro estudio se centra en las prácticas que incluyen: requerimientos, diseño, construcción, pruebas, mantenimiento, administración de la configuración, administración de la ingeniería, procesos, métodos y herramientas y calidad, de acuerdo a las recomendaciones e instrumento reportado en [GAR15].

### 3 Estudios sobre las prácticas de la ingeniería del *software* en la industria

Múltiples estudios se han realizado en distintos países para determinar el estado de las prácticas de la SE. Estos estudios se han realizado con el objetivo de entender esta industria, determinar oportunidades de mejora y fortalecer las colaboraciones entre la industria y la academia.

Garousi et al [GAR15] presentan un listado de encuestas realizadas en la industria sobre las prácticas de la SE abarcando todas sus áreas de conocimiento. Algunos de los estudios reportados posteriores al 2000 son: [GRO00] [HOA02] [AYT03] [CUS03] [DEN07] [FUJ07] [EGO09] [AYK09] [ZAI09] [EGO10] [SIN10] [KIR12] [VON12] [GAR15] [GAR16]. En el Cuadro 1 se presenta un resumen de estos estudios realizados del 2010 a la fecha.

**Cuadro 1 Resumen de encuestas sobre las prácticas de la SE**

<b>Ref.</b>	<b>País</b>	<b>Año</b>	<b>Participantes</b>	<b>Objetivo</b>
[EGO10]	Italia	2010	62 proyectos	Evaluar el efecto percibido de las prácticas de SE
[KIR12]	Nueva Zelanda	2012	195	Entender el uso de las prácticas de SE.
[VON12]	Holanda	2012	99	Aspectos de las prácticas de SE.
[GAR15]	Turquía	2015	202	Caracterización de las prácticas de SE
[GAR16]	Turquía	2016	202	Análisis de factores de influencia en las prácticas de SE
Este estudio	Costa Rica	2016	278	Caracterización de las prácticas de SE

Se debe considerar que este listado no es exhaustivo y no incluye encuestas que se realizan para sub temas específicos de la SE. Existen gran cantidad de estudios que analizan temas particulares de la SE como por ejemplo: prácticas relacionadas con la estimación [BOE00] [MOL04] [NAS06] [TRE08] [YAN08] [PAE12] [DAP16], administración y riesgos [CUT03] [ADD02] [NAS06], requerimientos [SOM97] [ZAI08] [TAH10], diseño [CUR88], desarrollo y productividad [CHE07] [SCH14], mejoramiento de procesos [TRA11] [SAN12] y pruebas y calidad de software [TOR03] [RUN03] [TEV04] [GER04] [MUR04] [TAI05] [SAE05] [RUN06] [TAI06] [GRI06] [REH07] [PAR08] [DIP09] [ENG10] [BOZ10] [GAR10] [CAU10] [LEE12] [GAR13] [DEA13] [DEA13] [NAN15] [DIA16].

Por restricciones de espacio, estos estudios no pueden ser detallados, sin embargo, la síntesis de la mayoría de ellos pueden ser encontrados en las siguientes revisiones de literatura [MOK03] [TRE08] [YAN08] [JAL12] [GAR13] [GAR15] [HEA15].

Nuestro estudio se basa en las prácticas generales del SWEBOK de acuerdo al diseño propuesto en [GAR15] que caracterizó las prácticas de ingeniería del *software* en la industria de Turquía.

## **4 Descripción del estudio**

En esta sección, describimos el diseño del estudio y la estrategia de ejecución. Este estudio realiza una encuesta para caracterizar las prácticas de la ingeniería del *software* (SE) utilizadas por la industria. El diseño del estudio es una replicación externa [SHU08] en la industria de software de Costa Rica y se basa en el estudio reportado en [GAR15]. El estudio incluye el análisis detallado de los datos recolectados, y su comparación con los estudios previos. Este estudio es descriptivo [LIN15].

### **4.1 Objetivo y preguntas de investigación**

El objetivo del estudio es caracterizar las prácticas de la SE en la industria de *software* costarricense. A partir del análisis de los datos se pretende entender la naturaleza y la madurez de las prácticas de las organizaciones de desarrollo que permita dirigir esfuerzos en el mejoramiento de su competitividad. Se espera que esto contribuya en el fortalecimiento de las colaboraciones entre la industria y la academia.

Los resultados del estudio analizan las prácticas de la SE en la industria, las técnicas, herramientas y métricas más comúnmente utilizadas. Además, detalla los principales desafíos en la industria que permiten identificar áreas de oportunidad en investigación para propiciar una cultura de mejoramiento continuo e investigación y desarrollo.

Con base en este objetivo se definen las siguientes preguntas de investigación. Estas se basan en [GAR15] y se detallan en el Cuadro 2.

**Cuadro 2 Preguntas de investigación**

<b>Id</b>	<b>Pregunta</b>	<b>Motivación</b>
RQ1	¿Cuál es el perfil profesional y las características de las organizaciones a la que pertenecen los participantes de la encuesta?	Identificar las áreas de experticia, el tamaño de las organizaciones, los tipos de servicios que brindan y así contextualizar las respuestas.
RQ2	¿Cuáles son las características del ciclo de vida de desarrollo de la industria de <i>software</i> de Costa Rica?	Conocer el nivel de involucramiento en cada una de las etapas y actividades del ciclo de vida de desarrollo del <i>software</i> .
RQ3	¿Qué tipos de prácticas de ingeniería de requerimientos, diseño de <i>software</i> , desarrollo, pruebas, mantenimiento, gestión de la configuración, administración de proyectos y gestión de la calidad son usadas por los profesionales?	Determinar las prácticas comunes que realizan los profesionales para cada una de las etapas del ciclo de vida de desarrollo del <i>software</i> y en la administración de sus procesos de <i>software</i> .
RQ4	¿Cuáles son las herramientas de ingeniería de <i>software</i> utilizadas por los profesionales?	Identificar el nivel de adopción de herramientas que soportan las actividades del desarrollo del <i>software</i> .
RQ5	¿Cuáles son los principales desafíos a que se enfrentan las organizaciones y cuál ha sido su nivel de investigación y colaboraciones con la academia?	Identificar los principales desafíos asociados a las prácticas de la SE y conocer el nivel de investigación y colaboraciones que realizan con la academia.

#### 4.2 Diseño de la encuesta

El diseño del cuestionario se basa en las recomendaciones y lineamientos reportados por [GRO09] [KIT02] [PFL01] [GAR15] [LIN15]. Para la selección de los participantes se utiliza un muestreo no probabilístico donde los participantes son contactados por medios electrónicos, donde participan de manera voluntaria y anónima. El muestreo por conveniencia, donde los participantes son contactados de acuerdo a las posibilidades de los investigadores es uno de los más utilizados en el área de la ingeniería del *software*, y los resultados deben ser interpretados con cautela dado el sesgo que puede introducirse por la muestra [SJO05].

En nuestro estudio, la unidad de análisis son profesionales relacionados con la ingeniería de *software* en la industria costarricense. Tal como en estudios previos, los resultados deben ser interpretados bajo esta condición y no pueden ser asociados a las organizaciones o sus proyectos de *software* [GAR15].

Se diseña un cuestionario electrónico basado en la estructura del SWEBOK [SWE04] [BOU14] y se revisan contra los reportados en estudios previos [GER04] [GAR10] [GAR13] [GAR15]. Nuestro estudio se limita a las áreas de conocimiento de los capítulos 1 al 10 de las prácticas del SWEBOK de acuerdo al diseño propuesto en [GAR15].

El cuestionario es piloteado con cinco profesionales de la industria costarricense para validar que el contenido y la terminología son entendibles. Con las recomendaciones recibidas se ajusta la versión final del cuestionario. El cuestionario cuenta con 46 preguntas basadas en las prácticas de la SE y directamente relacionadas con las preguntas de investigación del estudio.

La estructura general del cuestionario se presenta en el Cuadro 3. Dado la longitud del instrumento, en la medida de lo posible, el cuestionario fue diseñado con preguntas cerradas donde los participantes solo debían seleccionar de una lista de opciones.

**Cuadro 3 Estructura del cuestionario**

<b>RQ</b>	<b>Aspecto</b>	<b>Pregunta</b>	<b>Tipo</b>
1	Canal de contacto	Medio por el que fue contactado	Selección única.
	Perfil profesional	Roles	Selección múltiple.
		Experiencia	Numérico
		Educación	Selección única.
		Región	Selección única.
		Género	Selección única.
		Edad	Numérico
	Perfil de la organización	Idiomas	Selección múltiple.
		Sector	Selección única.
		Tamaño	Numérico
Proyectos	Organización	Selección múltiple.	
	Tipos	Selección única.	
	Clientes	Selección única.	
2	Ciclo de vida y prácticas organizacionales	Software desarrollado	Selección múltiple.
		Participación y dificultad	Likert
		Metodologías	Selección múltiple.
		Modelos de calidad	Selección única.
		Prácticas realizadas	Likert
3	Requerimientos	Prácticas realizadas	Likert
		Notaciones	Selección múltiple.
	Diseño	Prácticas realizadas	Likert
		Aspectos de diseño	Likert
	Desarrollo	Lenguajes	Selección múltiple.
		Prácticas realizadas	Likert
	Pruebas	Prácticas realizadas	Likert
		Fases de pruebas	Likert
		Tipos y niveles	Likert
		Diseño de pruebas	Selección múltiple.
Automatización		Likert	
Métricas		Likert	
Relación de testers desarrolladores		Selección única.	
Criterios de pruebas finalizadas		Likert	

	Mantenimiento	Actividades de mantenimiento	Likert
		Desafíos en tareas realizadas	Likert
	Administración de la configuración	Prácticas entrega y liberación realizadas	Likert
		Prácticas soporte realizadas	Likert
	Administración de proyectos	Prácticas realizadas	Likert
		Métodos de estimación	Likert
4	Herramientas y métodos	Herramientas utilizadas	Likert
		Mejores herramientas	Abierta
	Aseguramiento de la calidad	Prácticas realizadas	Likert
5	Investigación y colaboración con la academia	Unidad de investigación	Sí/No
		Desafíos para su organización y opinión sobre las prácticas de la SE	Abierta
		Frecuencia de interacción con investigadores del área	Selección única.
		Frecuencia de lectura de artículos científicos	Selección única.
		Valor de la literatura científica	Abierta

### 4.3 Ejecución

El cuestionario electrónico fue implementado en la herramienta *LimeSurvey* ([www.limesurvey.org](http://www.limesurvey.org)) y se mantuvo en línea 4 meses, desde julio a octubre del 2016. Todos los participantes fueron invitados a participar de manera anónima y voluntaria por correos de invitación enviados por medio de distintas instituciones relacionadas con la industria de *software* en Costa Rica. La muestra incluye profesionales relacionados con la ingeniería del *software* en el país (servicios de desarrollo y pruebas de *software* tanto en el sector público como privado). Dados los distintos perfiles, se les indicó a los participantes que respondieran las secciones que consideran de acuerdo a sus áreas de experticia.

Similar a experiencias reportadas en estudios anteriores [GAR13] [GAR15], algunos participantes dejaron preguntas sin responder y otros abandonaron el cuestionario sin terminarlo. Todas las respuestas parciales fueron consideradas para el análisis de resultados puesto que son consideradas valiosas.

Después del pre-procesamiento de datos, se analizan las respuestas de 278 profesionales de ingeniería de software trabajando en organizaciones localizadas en Costa Rica. El detalle del número de participantes de la encuesta se detalla en el Cuadro 4.

**Cuadro 4 Participantes**

<b>Cantidad de profesionales</b>	
Contactados	No es posible determinar.
Ingresan al cuestionario	313
Responden el cuestionario	278
No responden el cuestionario	35

El detalle de los medios de contacto más efectivos se detalla en el Cuadro 5, el medio más efectivo de contacto fueron las Universidades Públicas seguido por las Instituciones del Sector Público.

**Cuadro 5 Canales de contacto**

<b>Institución</b>	<b>n</b>	<b>%</b>
Universidades Públicas	192	69.1%
Instituciones sector público	72	25.9%
Cámaras de Industria	4	1.4%
Universidades Privadas	2	0.7%
Colegio de Profesionales	1	0.4%
Otros	7	2.5%
<b>Total</b>	<b>278</b>	<b>100%</b>

En el Cuadro 6 se detallan la tasa de respuestas por sección de la encuesta. En promedio responden de manera completa todas las secciones de la encuesta 120 participantes para una tasa de respuesta del 43%. Este comportamiento es similar al reportado en estudios previos en el área [PUN03] [GAR10] [GAR13] [GAR15].

Los participantes que fueron contactados por medio de las universidades tienen una tasa de finalización de respuestas del 40% y los de las instituciones públicas del 61%. Los desarrolladores e ingenieros de requerimientos tienen una tasa de finalización de entre 40-43%, los analistas o ingenieros de *software*, arquitectos y diseñadores del 33-34%, los analistas de negocio y líderes de proyecto del 30%, los administradores de proyectos del 25% y los gerentes de alto nivel (CEO, CIO, CFO) e ingenieros de calidad del 12-13%. Los profesionales que tienen alguna relación con labores docentes tienen una tasa de finalización de respuestas del 38% y los relacionados con labores de investigación del 10%.

Analizando las tendencias de preguntas no respondidas podemos observar que la mayor cantidad de preguntas sin contestar se relacionan con las prácticas específicas en la SE. Dos factores pueden influenciar esta tendencia son: (1) la cantidad de preguntas del cuestionario y (2) el que algunos de los participantes proveen respuestas de alto nivel, pero son reservados acerca de las prácticas específicas en sus organizaciones.

En la invitación a la encuesta se les indicó a los participantes que las encuestas eran anónimas y los resultados publicados serían mostrados de manera agregada. Se espera que en futuros cuestionarios los participantes se sientan motivados a responder la totalidad el cuestionario al observar que los datos reportados no afectan la privacidad de sus organizaciones.

**Cuadro 6 Respuestas por sección del cuestionario**

<b>Sección del cuestionario</b>	<b>Fin</b>	<b>Acum.</b>	<b>%</b>	<b>Tasa</b>
1 Consentimiento informado	3	278	1%	100%
2 Perfil profesional	17	275	6%	99%
3 Perfil de la organización	6	258	2%	93%
4 Proyectos de desarrollo de software	132	252	47%	91%
5 Prácticas organizacionales	120	120	43%	43%

## **5 Resultados**

En esta sección presentamos los resultados de la encuesta y analizamos la información para cada una de las preguntas de investigación.

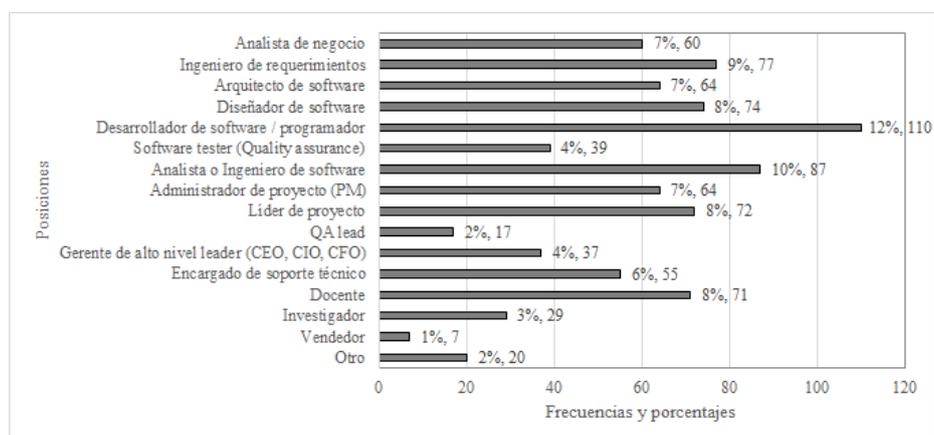
### **5.1 Perfiles y datos demográficos**

La RQ1 busca determinar los perfiles profesionales y las características de las organizaciones y los proyectos a la que pertenecen los participantes de la encuesta.

#### **5.1.1 Perfil de los participantes**

**Posiciones de los participantes.** Las posiciones reportadas por los 278 participantes se muestran en la Figura 1 y el Cuadro 7. La pregunta fue diseñada para permitir selección múltiple puesto que conocemos que muchos de los profesionales de la industria desempeñan múltiples roles.

La mayoría de las respuestas son de los perfiles de desarrollador de software y programador e ingeniero de software, seguido por perfiles relacionados con la práctica ingenieril tales como requerimientos y diseño. Además encontramos líderes de proyectos que se consideran puestos técnicos y profesionales relacionados con puestos docentes. Cabe resaltar que la mayoría de los profesionales en puestos docentes combinan esta labor con la práctica profesional (80%) y en todos los casos forman profesionales relacionados con la SE. La opinión de los participantes desde las diferentes perspectivas profesionales y puntos de vista es importante para el estudio.



**Figura 1 Cantidad de participantes por posición**

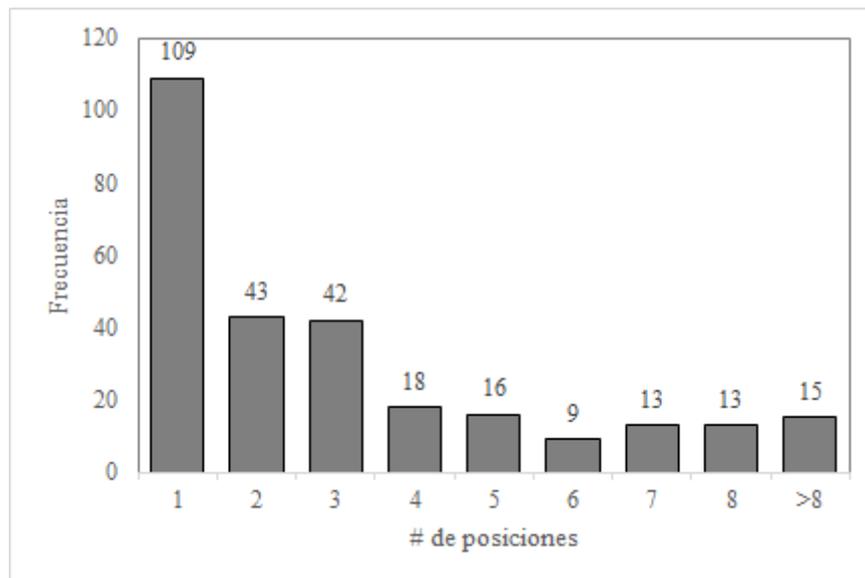
En nuestro estudio, las posiciones profesionales reportadas por los participantes son similares a las reportadas en [GAR15], por lo que es interesante a partir de la recolección de los datos comparar algunas de las tendencias obtenidas.

**Cuadro 7 Cantidad de participantes por posición**

Posición	n	%
Desarrollador de software / programador	110	12%
Analista o Ingeniero de software	87	10%
Ingeniero de requerimientos	77	9%
Diseñador de software	74	8%
Líder de proyecto	72	8%
Docente	71	8%
Arquitecto de software	64	7%
Administrador de proyecto (PM)	64	7%
Analista de negocio	60	7%
Encargado de soporte técnico	55	6%
Software tester ("Quality assurance")	39	4%
Gerente de alto nivel leader (CEO, CIO, CFO)	37	4%
Investigador	29	3%
"QA lead"	17	2%
Vendedor	7	1%
Otro	20	2%
<b>Total</b>	<b>883</b>	<b>100%</b>

La Figura 2 muestra las frecuencias y el Cuadro 8 las frecuencias y porcentajes de los casos en que un profesional desempeña una o más posiciones. El 39% de los participantes reportan solo una posición en sus organizaciones. Sin embargo, 84 participan-

tes (30%) reportan desempeñarse en 4 o más posiciones. La mayoría de estos profesionales se encuentran trabajando en organizaciones nacionales grandes (501+ empleados) y en el que su giro de negocio no es el del desarrollo de software.



**Figura 2** Cantidad de posiciones desempeñadas por cada participante

En comparación con los estudios previos [GAR13] [GAR15] podemos observar que las tendencias son similares, en que en los tres estudios la mayoría de los profesionales se desempeña en una única posición seguida por dos posiciones.

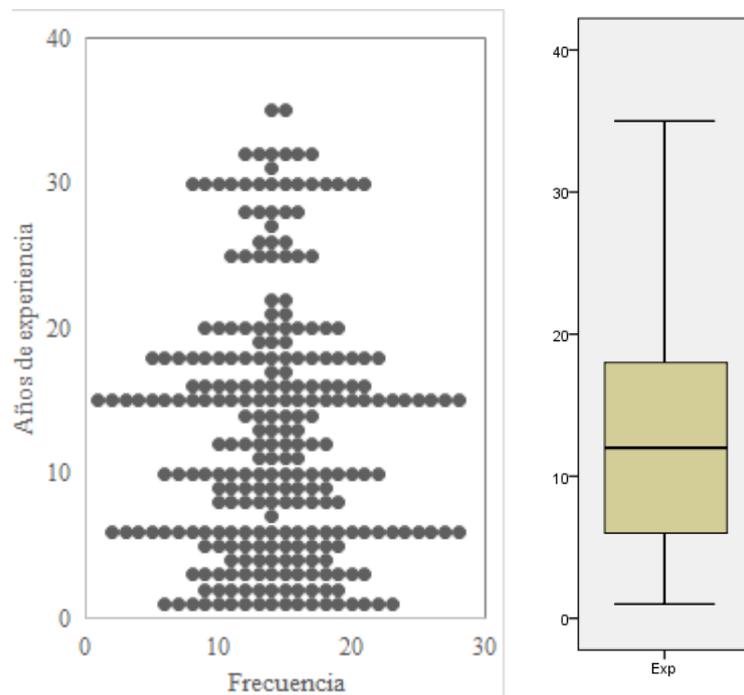
**Cuadro 8** Cantidad de posiciones desempeñadas por cada participante

Posiciones	n	%
1	109	39%
2	43	15%
3	42	15%
4	18	6%
5	16	6%
6	9	3%
7	13	5%
8	13	5%
>8	15	5%
<b>Total</b>	<b>278</b>	<b>100%</b>

**Experiencia laboral.** Los años de experiencia de los participantes se muestran en la Figura 3. El promedio de experiencia es de 12.9 años y la media de 12 años. La experiencia de los participantes con igual o menos de 10 años es 47%. Alrededor del 23%

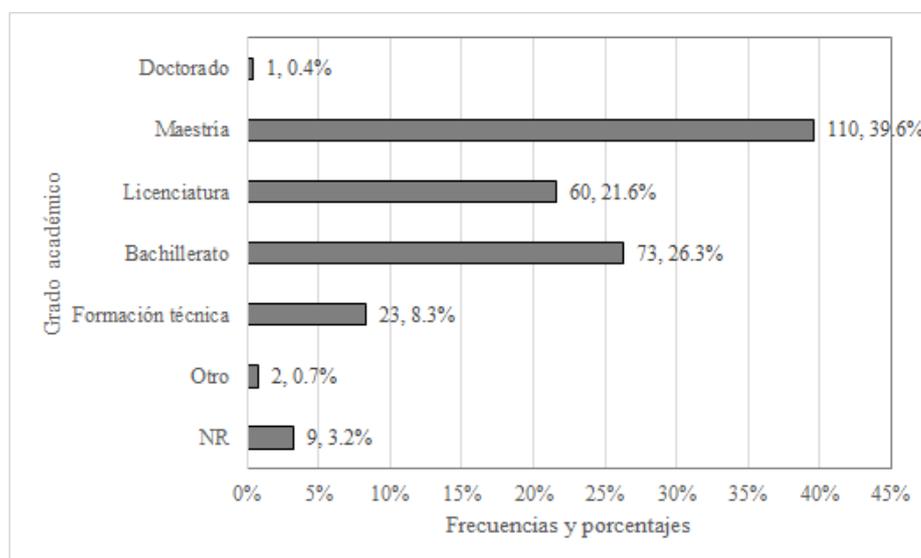
de los participantes tiene 5 o menos años de experiencia, el 24% entre 5 y 10 años de experiencia y el 53% más de 10 años. La variación de experiencia es considerada valiosa para nuestro estudio. El promedio de edad de los participantes es de 37 años (P25=30, P50=37, P75=44), el 76% de los participantes son masculinos y el 21% femeninas.

En comparación con los estudios previos, podemos observar que las tendencias de años de experiencia es mayor en nuestro estudio que en [GAR13] (Pm=7.8, Md=6) y [GAR15] (Pm=9.5, Md=8). Las tendencias muestran que la participación de profesionales jóvenes en nuestra encuesta es menor que la de los de la contraparte Canadiense y Turca.



**Figura 3 Años de experiencia**

**Perfil académico.** Para conocer los antecedentes académicos de los participantes se consulta su máximo grado académico, si persiguen un grado académico en este momento y su área de estudio. La Figura 4 y el Cuadro 9 muestran el grado académico más alto obtenido por los participantes. El 40% de los participantes cuentan con una maestría, el 22% con una licenciatura y el 26% con el bachillerato. Además, el 49% de los participantes se encuentra en este momento estudiando un grado superior o una actualización técnica, dentro de los cuales el 27% se encuentran estudiando una maestría, el 36% una licenciatura y el 11% un título de especialización técnica.



**Figura 4 Grado académico más alto obtenido**

En comparación con los estudios previos [GAR13] [GAR15], podemos observar que las tendencias se invierten entre maestrías y bachilleratos, sin embargo si se unen las licenciaturas con los bachilleratos se muestran proporciones similares.

**Cuadro 9 Grado académico más alto obtenido**

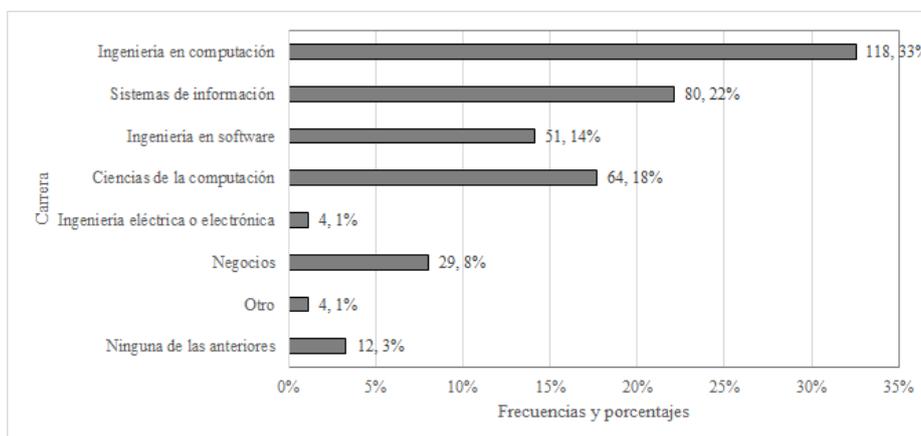
Grado	n	%
Doctorado	1	0.4%
Maestría	110	39.6%
Licenciatura	60	21.6%
Bachillerato	73	26.3%
Formación técnica	23	8.3%
Otro	2	0.7%
NR	9	3.2%
<b>Total</b>	<b>278</b>	<b>100.0%</b>

**Formación universitaria.** Para determinar las habilidades de los participantes, se les consulta sobre el nombre de la carrera de su formación universitaria. Los resultados se detallan en la Figura 5 y el Cuadro 10. La pregunta fue diseñada para permitir múltiple selección puesto que muchos de los profesionales estudiaron múltiples carreras.

Ingeniería en computación (33%), sistemas de información (22%), ciencias de la computación (18%), e ingeniería del software (14%) son las principales carreras de los profesionales. La mayoría de los entrevistados estudió en las universidades públicas costarricenses (66%), tal como se muestra en el Cuadro 11.

En comparación con los estudios previos [GAR13] [GAR15] podemos observar que la tendencia se mantiene para carreras relacionadas con la computación pero no para

carreras relacionadas con negocios o ingeniería electrónica en la cual nuestro estudio cuenta con menor número de participantes.



**Figura 5 Carrera de estudio**

**Cuadro 10 Carrera de estudio**

<b>Carrera</b>	<b>n</b>	<b>%</b>
Ingeniería en computación	118	33%
Sistemas de información	80	22%
Ingeniería en software	51	14%
Ciencias de la computación	64	18%
Ingeniería eléctrica o electrónica	4	1%
Negocios	29	8%
Otro	4	1%
Ninguna de las anteriores	12	3%
<b>Total</b>	<b>362</b>	<b>100%</b>

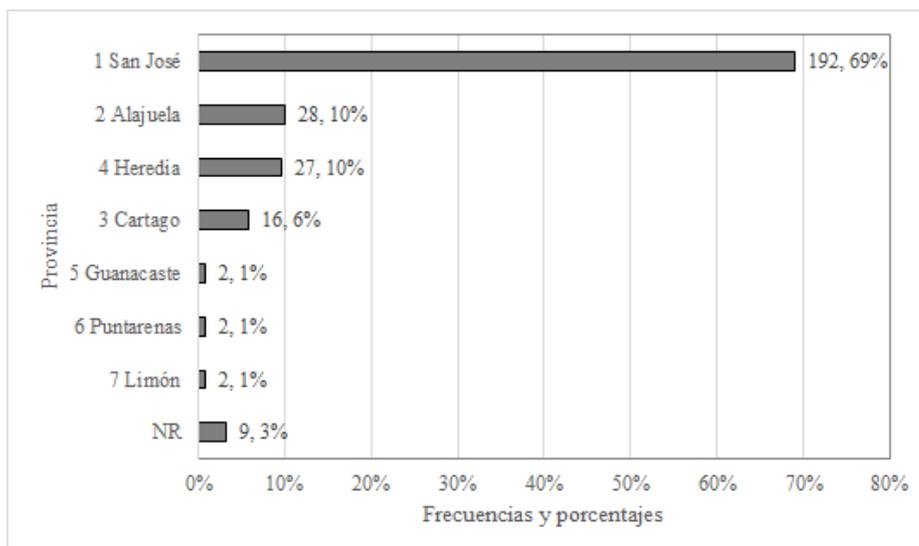
**Cuadro 11 Universidades**

<b>Universidad</b>	<b>%</b>
UCR Universidad de Costa Rica	21.5%
ITCR Instituto Tecnológico de Costa Rica	16.9%
UNA Universidad Nacional	9.7%
UNED Universidad Estatal a Distancia	11.8%
UTN Universidad Técnica Nacional	4.4%
Privadas Universidades Privadas	26.0%
Otras Otras universidades	8.3%

**Ubicación geográfica.** La ubicación geográfica de los participantes se muestra en la Figura 6 y el Cuadro 12.

San José (69%), Alajuela (10%), Heredia (10%) y Cartago (6%) son las provincias donde más participantes están localizados. Estas provincias se encuentran en la gran

área metropolitana donde se localizan la mayoría de organizaciones de tecnología del país. Al consultar por la provincia de origen, de igual forma la mayoría de los participantes se ubican en la gran área metropolitana, San José (37%), Alajuela (19%), Heredia (14%) y Cartago (15%).



**Figura 6 Provincia donde trabaja**

**Cuadro 12 Provincia donde trabaja**

<b>Provincia</b>	<b>n</b>	<b>%</b>
1 San José	192	69%
2 Alajuela	28	10%
4 Heredia	27	10%
3 Cartago	16	6%
5 Guanacaste	2	1%
6 Puntarenas	2	1%
7 Limón	2	1%
NR	9	3%
<b>Total</b>	<b>278</b>	<b>100%</b>

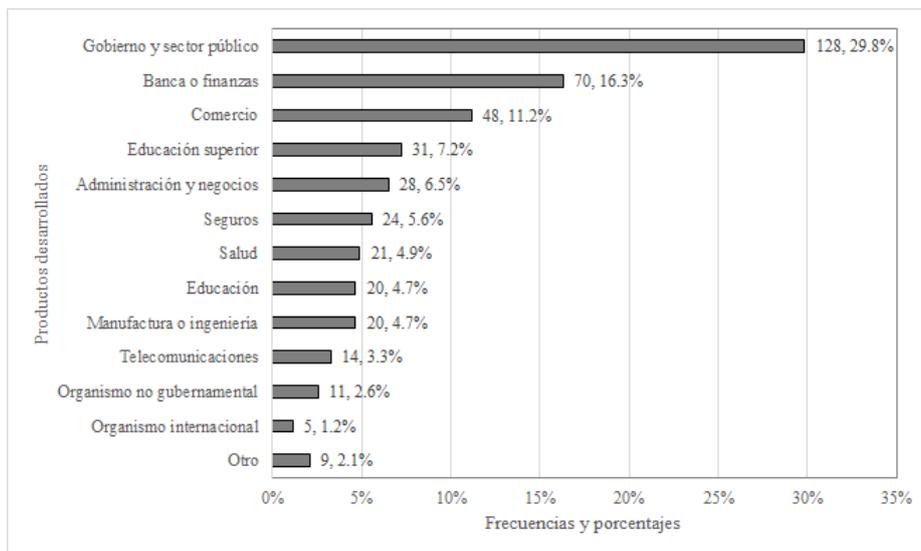
### 5.1.2 Perfil de las organizaciones

**Productos desarrollados.** La Figura 7 y el Cuadro 13 presentan los tipos de productos desarrollados por las organizaciones de software.

El sector objetivo de los productos de software son en su mayoría para el gobierno y sector público (30%), la banca y finanzas (16%) y comercio electrónico en general (11%). Como se observa, se presenta una buena combinación de respuestas en diferentes dominios.

La principal razón de esta tendencia es que la mayoría de los participantes pertenecen al sector gobierno (24%) y a la industria de desarrollo de software a la medida que ofrece múltiples servicios al sector público y privado (14%). Finalmente, muchos de los participantes pertenecen al sector de banca y finanzas (11%).

La mayoría de las organizaciones de los participantes son de capital nacional (67%), el 12% de las organizaciones son de capital extranjero y el 10% de ambos.



**Figura 7 Tipo de productos desarrollados**

En comparación con los estudios previos [GAR15] podemos observar que la industria en Costa Rica es distinta a la de Turquía, puesto que nuestro país no se desarrolla la industria militar y de defensa. Además, la cantidad de participantes de la industria de telecomunicaciones en nuestra encuesta es solo del 3.3%.

**Cuadro 13 Tipo de productos desarrollados**

<b>Tipo de productos desarrollados</b>	<b>n</b>	<b>%</b>
Gobierno y sector público	128	29.8%
Banca o finanzas	70	16.3%
Comercio	48	11.2%
Educación superior	31	7.2%
Administración y negocios	28	6.5%
Seguros	24	5.6%
Salud	21	4.9%
Manufactura o ingeniería	20	4.7%
Educación	20	4.7%
Telecomunicaciones	14	3.3%
Organismo no gubernamental	11	2.6%
Organismo internacional	5	1.2%
Otro	9	2.1%
<b>Total</b>	<b>429</b>	<b>100.0%</b>

**Tamaño de las organizaciones.** La Figura 8 y el Cuadro 14 presentan la cantidad de participantes por tamaño de la organización donde trabajan, se presentan los datos de los participantes que contestaron el cuestionario de manera completa.

Más del 64% de los participantes pertenecen a organizaciones grandes (501+ empleados) y el 57% de los participantes indican que su organización cuenta con una oficina dedicada a la administración de proyectos.

En comparación con los estudios previos, podemos observar que la tendencia es similar a [GAR15] donde la mayoría de los participantes son de organizaciones grandes.

Además, los tipos de clientes para los que trabajan las organizaciones son clientes internos (27%), clientes externos (13%) y en su mayoría para ambos tipos (42%).

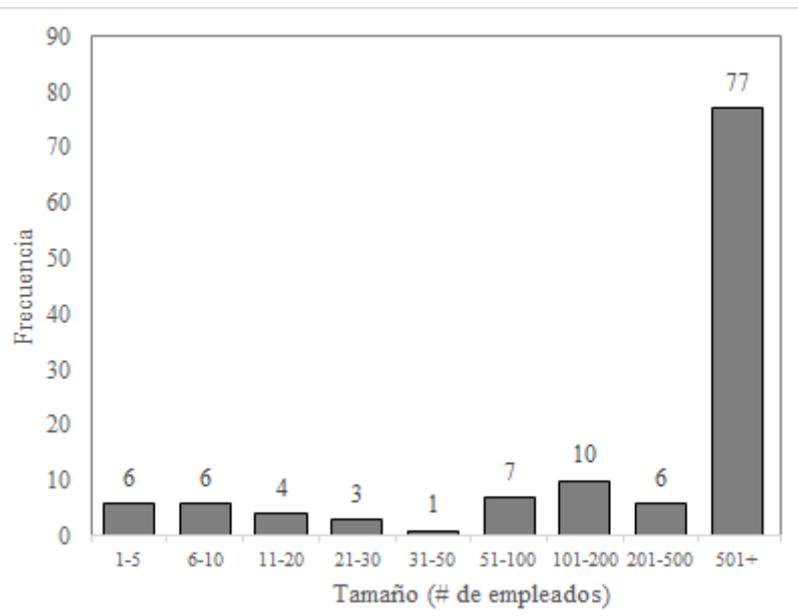


Figura 8 Tamaño de las organizaciones

Cuadro 14 Tamaño de las organizaciones

Tamaño	n	%
1-5	6	5%
6-10	6	5%
11-20	4	3%
21-30	3	3%
31-50	1	1%
51-100	7	6%
101-200	10	8%
201-500	6	5%
501+	77	64%
<b>Total</b>	<b>120</b>	<b>100.0%</b>

### 5.1.3 Perfil de los proyectos de *software*

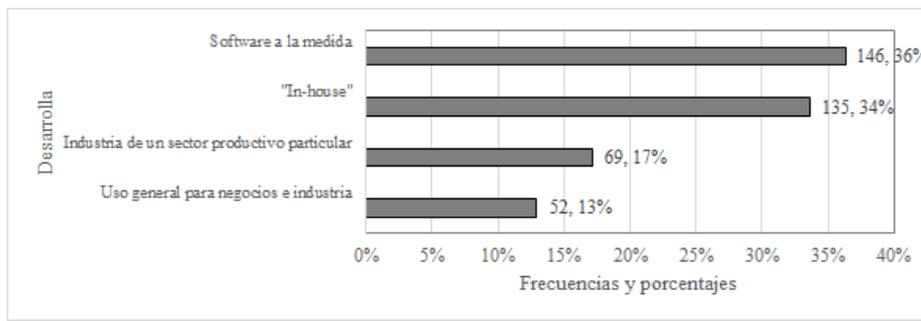
**Tipo de industria.** Para el análisis de los perfiles de los proyectos de software se consulta sobre el tipo de industria al que se destinan los productos que se producen.

La Figura 9 y el Cuadro 15 presentan los tipos de productos desarrollados por las organizaciones de software. Los proyectos son en su mayoría productos de software a la medida a partir de las necesidades particulares de los clientes (36%). Además, gran

cantidad del software se desarrolla "In-house" (34%). El desarrollo de paquetes de software para industrias particulares es del 17% y el de uso general para negocios es de 13% y representa un tipo de industria objetivo importante.

La pregunta permite múltiples respuestas, donde el 50% indican una sola opción, el 30% dos opciones, el 13% tres opciones y el 7% las cuatro posibles.

En comparación con estudios previos [GAR15] podemos observar una tendencia similar hacia el desarrollo basado en las necesidades particulares de los clientes y desarrollo de software "In-house". En el caso de nuestro estudio, 50% de los participantes indican que desarrollan para más de una industria, contrario al 22% indicado en la contraparte. El 67% de los participantes de nuestro estudio indican que los proyectos que desarrollan son productos de software a la medida a partir de las necesidades particulares de los clientes y a la vez que es software desarrollado "In-house".



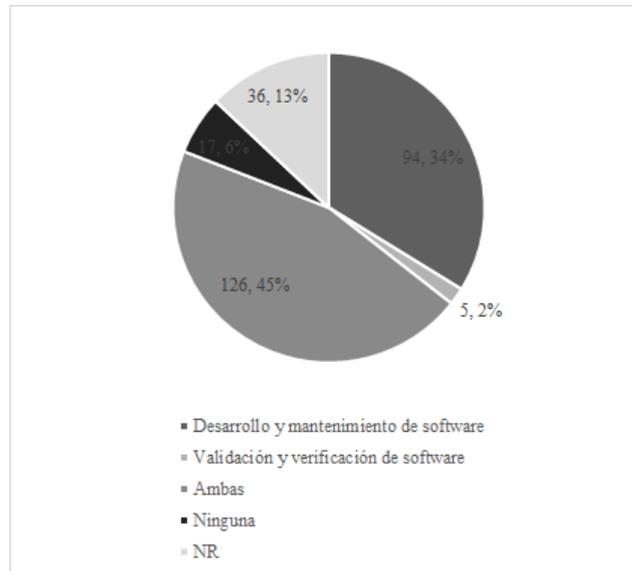
**Figura 9 Tipos de industria para los productos**

**Cuadro 15 Tipos de industria para los productos**

Tipos de productos	n	%
Software a la medida a partir de las necesidades de los clientes	146	36%
Software "In-house"	135	34%
Paquetes de software para una industria de un sector particular	69	17%
Paquete de software de uso general para negocios e industria	52	13%
<b>Total</b>	<b>402</b>	<b>100%</b>

**Tipo de proyectos.** La Figura 10 y el Cuadro 16 presentan los tipos de proyectos que se llevan a cabo en las organizaciones.

El 45% de los participantes indican que realizan proyectos de desarrollo, validación y verificación, y mantenimiento de software, el 34% llevan a cabo específicamente proyectos de desarrollo y mantenimiento y el 2% indica que realizan específicamente proyectos de validación y verificación.



**Figura 10 Tipo de proyectos**

**Cuadro 16 Tipo de proyectos**

<b>Tipo de proyectos</b>	<b>n</b>	<b>%</b>
Desarrollo y mantenimiento de software	94	34%
Validación y verificación de software	5	2%
Ambas	126	45%
Ninguna	17	6%
NR	36	13%
<b>Total</b>	<b>278</b>	<b>100%</b>

## 5.2 Ciclo de vida y prácticas organizacionales

La RQ2 busca determinar las características del ciclo de vida de desarrollo, los procesos generales de la ingeniería del *software* y el mejoramiento de procesos. Para esto se consulta sobre el nivel de participación y desafío experimentado durante el ciclo de vida y las principales prácticas organizacionales relacionadas con las metodologías y modelos de calidad.

### 5.2.1 Nivel de participación en las fases de los proyectos de *software*

El nivel de involucramiento de los participantes en las diferentes fases y actividades del ciclo de vida de desarrollo se muestra en la Figura 11 y el Cuadro 17. Los participantes son consultados por fase de desarrollo en una escala de Likert con las siguientes opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Son consultados sobre su participación en las fases de requerimientos, diseño, desarrollo, pruebas y mantenimiento; además de las actividades de administración de la configuración, administración de proyectos, administración de procesos, aseguramiento de la calidad y documentación técnica.

Los participantes reportan participación activa en las fases de requerimientos, diseño, desarrollo, pruebas y mantenimiento. Además reportan su participación en el proceso de administración de proyectos. Las fases con menor nivel de participación son las de administración de la configuración, definición, evaluación y medición de los procesos de ingeniería del *software*, aseguramiento de la calidad y documentación técnica.

Al considerar sólo los participantes que reportan trabajar como desarrolladores y programadores la tendencia es similar, donde para requerimientos, diseño, desarrollo, pruebas y mantenimiento, muestra un ligero predominio, las fases de desarrollo y mantenimiento de *software*.

En comparación con estudios previos [GAR15] podemos observar una tendencia similar hacia la fase de desarrollo y la baja participación en actividades de administración de la configuración, definición, evaluación y medición de los procesos de ingeniería del *software*, aseguramiento de la calidad y documentación técnica. Contrario a nuestro estudio los participantes en [GAR15] no reportan alta participación en procesos de administración de proyectos.

Es necesario más análisis sobre los factores que influyen en el nivel de participación en las distintas fases y actividades del ciclo de vida.

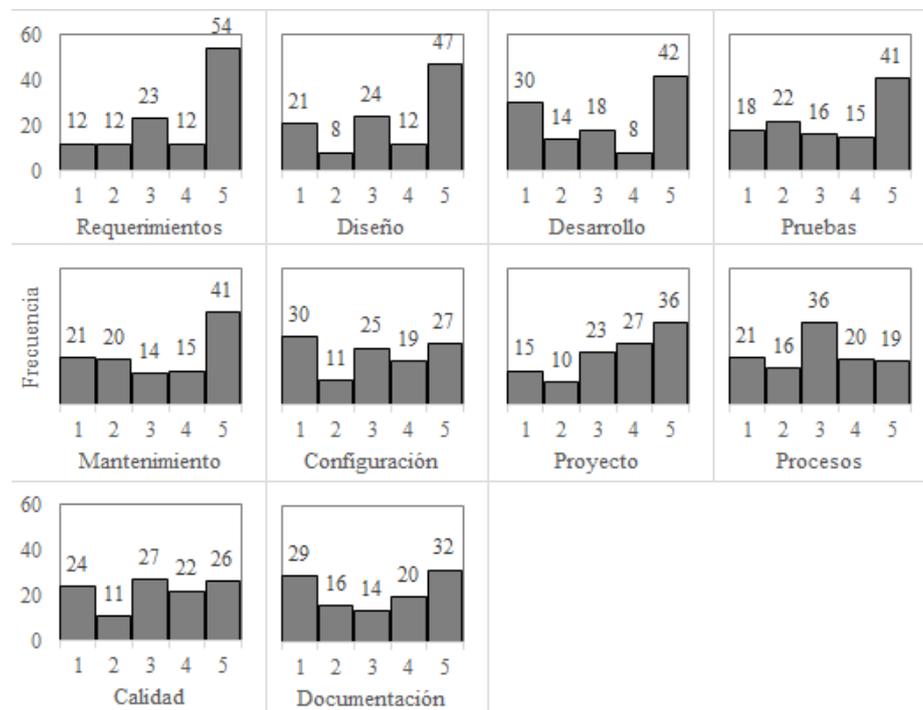


Figura 11 Frecuencia de respuestas por nivel de participación

**Cuadro 17 Nivel de participación en las fases del ciclo de vida de desarrollo**

Ciclo de vida de desarrollo	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Requerimientos	11	11	20	11	<b>48</b>	<b>3.74</b>	<b>4</b>	1.42
Diseño	19	7	21	11	<b>42</b>	3.50	<b>4</b>	1.54
Desarrollo	<b>27</b>	13	16	7	<b>38</b>	3.16	3	<b>1.66</b>
Pruebas	16	20	14	13	<b>37</b>	3.35	<b>4</b>	1.53
Mantenimiento	19	18	13	14	<b>37</b>	3.32	<b>4</b>	1.57
Adm. de la configuración	27	10	22	17	<b>24</b>	3.02	3	1.52
Administración del proyecto	14	9	21	<b>24</b>	<b>32</b>	<b>3.53</b>	<b>4</b>	1.38
Procesos: definir, evaluar, medir	19	14	<b>32</b>	18	17	3.00	3	1.33
Aseguramiento de la calidad	22	10	<b>25</b>	20	<b>24</b>	3.14	3	1.46
Documentación técnica	<b>26</b>	14	13	18	<b>29</b>	3.09	3	<b>1.59</b>

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

### 5.2.2 Nivel de desafío experimentado en las distintas fases y actividades

El nivel de desafío experimentado por los participantes en las diferentes fases del ciclo de vida de desarrollo se muestra en la Figura 12 y el Cuadro 18. Los participantes son consultados por fase y actividad en una escala de Likert con las siguientes opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Son consultados sobre su nivel de desafío para las fases de requerimientos, diseño, desarrollo, pruebas y mantenimiento; además en las actividades de administración de la configuración, administración de proyectos, administración de procesos, aseguramiento de la calidad y documentación técnica. Finalmente, se consulta sobre los desafíos relacionados con las herramientas de la ingeniería del software, y la comunicación entre los miembros del equipo, la comunicación con la alta administración y con los usuarios finales.

Similar a los resultados reportados en [GAR15], los participantes reportan un nivel de desafío levemente superior para las actividades de aseguramiento de calidad y comunicación con la alta administración y con los usuarios finales. Dentro de las fases de ciclo de vida de desarrollo la etapa reportada como más desafiante es la de pruebas.

Es necesario un análisis más profundo sobre los factores que influyen en el nivel de desafío en las distintas fases y actividades del ciclo de vida para determinar las necesidades de capacitación en cada una de ellas.

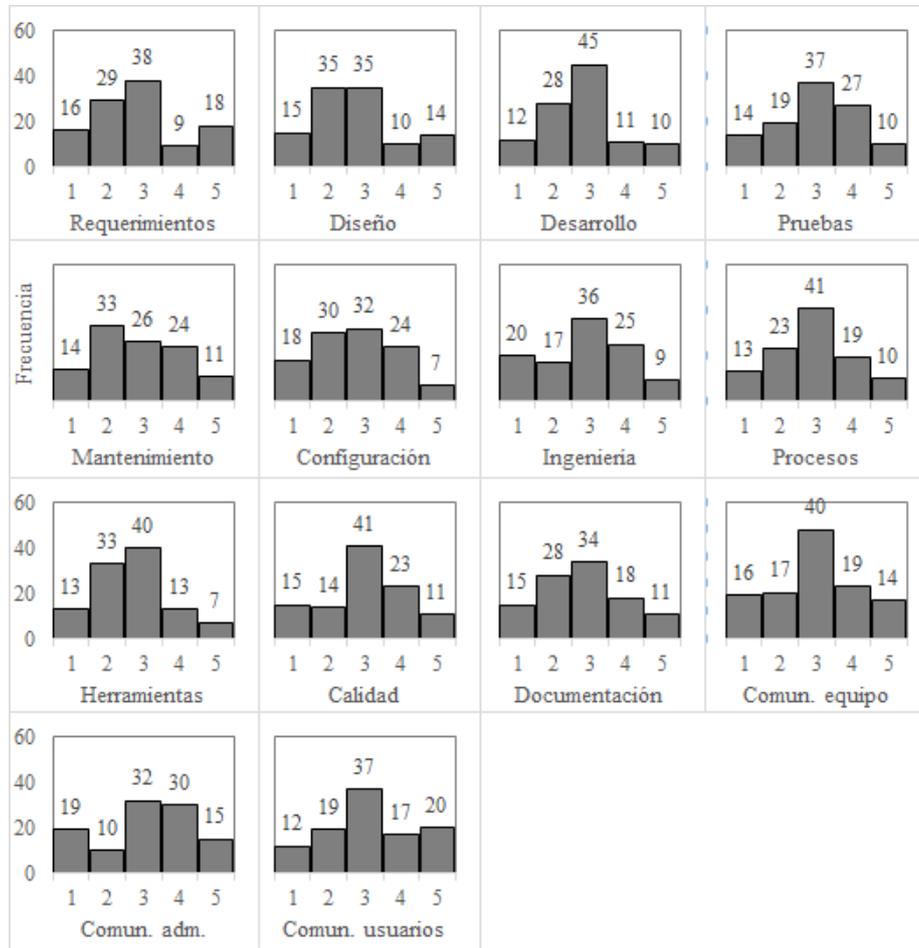


Figura 12 Frecuencia de respuestas por nivel de desafío experimentado

Cuadro 18 Nivel de desafío experimentado

Ciclo de vida de desarrollo	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Requerimientos	15	26	<b>35</b>	8	16	2.85	3	<b>1.25</b>
Diseño	14	<b>32</b>	<b>32</b>	9	13	2.75	3	1.20
Desarrollo	11	26	<b>42</b>	10	9	2.80	3	1.08
Pruebas	13	18	<b>35</b>	25	9	3.00	3	1.16
Mantenimiento	13	<b>31</b>	24	22	10	2.86	3	1.20
Adm. de la configuración	16	27	<b>29</b>	22	6	2.75	3	1.16
Administración del proyecto	19	16	<b>34</b>	23	8	2.87	3	1.21
Administración de procesos	12	22	<b>39</b>	18	9	2.91	3	1.13
Herramientas de ingeniería	12	<b>31</b>	<b>38</b>	12	7	2.70	3	1.05
Aseguramiento de la calidad	14	13	<b>39</b>	22	11	<b>3.01</b>	3	1.17
Documentación técnica	14	26	<b>32</b>	17	10	2.83	3	1.18
Comunicación del equipo	15	16	<b>38</b>	18	13	2.98	3	1.22
Comunicación alta gerencia	18	9	<b>30</b>	<b>28</b>	14	<b>3.11</b>	3	<b>1.29</b>
Comunicación con usuarios finales	11	18	<b>35</b>	16	19	<b>3.13</b>	3	<b>1.25</b>

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

### 5.2.3 Metodologías de desarrollo de *software*

La Figura 13 y el Cuadro 19 presentan las metodologías de desarrollo utilizadas por los participantes. La pregunta fue diseñada para permitir selección múltiple. Los participantes son consultados sobre el uso de metodologías ágiles, metodologías tradicionales y otras.

Las metodologías ágiles son las más utilizadas por los profesionales (46.7%). La metodología ágil más popular es *Scrum* con un 24.5%, el 19.4% indican seguir las prácticas ágiles sin adoptar un método específico en particular y finalmente tenemos al XP con un 2.8%. Metodologías tradicionales como cascada (19.9%), prototipos (19.0%), y en menos medida desarrollo incremental (10.2%) y espiral (3.7%) aún se mantienen vigentes en la industria. Finalmente, la metodología menos utilizada es la de "*Product-line development*" con un 0.5%. El 85 % de los profesionales usan o una metodología ágil o la metodología de cascada pero no ambas, esto indica que si una organización utiliza metodologías ágiles no utiliza cascada y viceversa.

Es necesario un análisis más profundo para determinar los factores que influyen el uso de las metodologías de acuerdo a las características de los proyectos y las organizaciones.

Los resultados son similares a los reportados en [GAR15] con respecto a la utilización de las metodologías ágiles, sin embargo la utilización de la metodología de cascada es mucho mayor a nuestros resultados (53%) al igual que la de "*Product-line development*" (12%). Estos resultados pueden ser explicados por el tipo de industria reportado en [GAR15] donde la mayoría de las organizaciones reportadas es del área militar y defensa [GAR16].

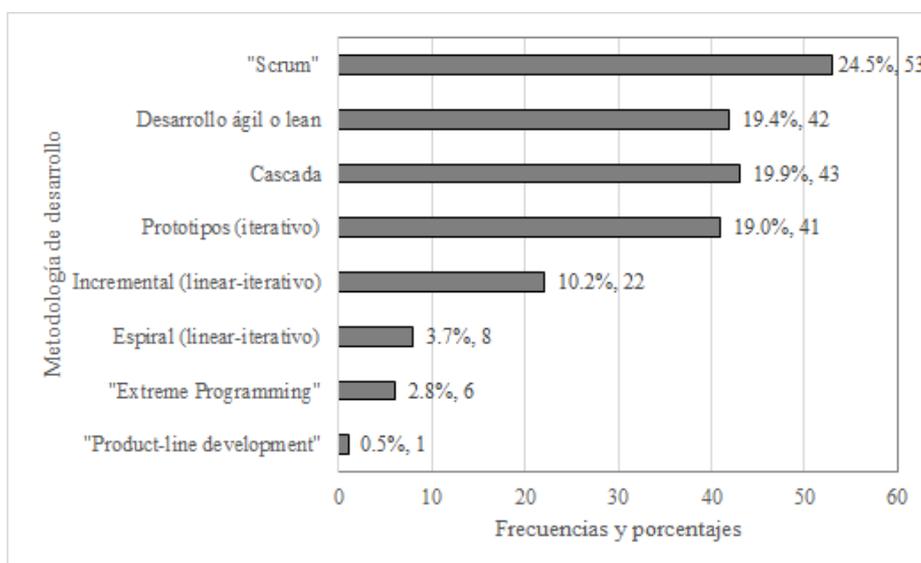


Figura 13 Metodologías de desarrollo

Cuadro 19 Metodologías de desarrollo

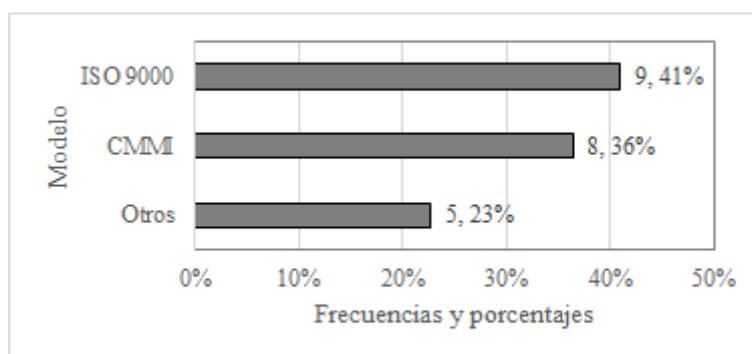
Metodologías	n	%
"Scrum"	53	24.5%
Desarrollo ágil o lean	42	19.4%
Cascada	43	19.9%
Prototipos (iterativo)	41	19.0%
Incremental (linear-iterativo)	22	10.2%
Espiral (linear-iterativo)	8	3.7%
"Extreme Programming"	6	2.8%
"Product-line development"	1	0.5%
<b>Total</b>	<b>216</b>	<b>100.0%</b>

#### 5.2.4 Modelos de mejoramiento de procesos

La Figura 14 y el Cuadro 20 presentan los modelos de mejoramiento de procesos adoptados por las organizaciones. Para esta pregunta solo respondieron 22 participantes de los cuales el 41% utilizan el estándar ISO 9000 ("*Quality management systems*") y el 36% utilizan el CMMI ("*The Capability Maturity Model Integration*"). El 23% de los participantes mencionan el uso de otros modelos tales como COBIT, ITIL, y otros estándares como el ISO 27001, ISO 42010 e ISO 19510.

Los resultados son similares en cuanto a porcentajes entre los modelos reportados en [GAR15], pero la adopción de los modelos de mejoramiento de procesos en la industria costarricense es mucho menor. En [CAM14] se reporta que solo el 13.2% de las orga-

nizaciones relacionadas con tecnologías de información han adoptado formalmente modelos o estándares de calidad, de las cuales el 48.3% adoptaron el ISO 9000 y el 24.1 el CMMI. Además indican que el 21.5% de las organizaciones planean incorporar algún modelo de calidad.



**Figura 14 Modelos de mejora de procesos**

**Cuadro 20 Modelos de mejora de procesos**

Modelos	n	%
ISO 9000 (" <i>Quality management systems</i> ")	9	41%
CMMI (" <i>The Capability Maturity Model Integration</i> ")	8	36%
Otros	5	23%
<b>Total</b>	<b>22</b>	<b>100%</b>

### 5.2.5 Prácticas relacionadas con procesos

Esta pregunta investiga acerca de las prácticas llevadas a cabo por las organizaciones en relación a los procesos. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si basan sus tareas en un estándar de procesos de un manual de calidad, si las tareas definidas por el equipo de administración de proyectos basadas en un manual de calidad, si realizan un monitoreo y evaluación sistemática del proceso de desarrollo usando métricas, o si realizan un proceso sistemático para el aprendizaje de proyectos anteriores que les permita optimizar el rendimiento de los proyectos futuros.

La Figura 15 y el Cuadro 21 presentan las frecuencias y porcentajes de cada enfoque particular. Aunque la mayoría de las organizaciones no reportan la adopción formal de modelos o estándares, la tendencia se inclina levemente a la aplicación de tareas basadas en un estándar de procesos de un manual de calidad. De manera similar reportan el uso de tareas definidas por el equipo de administración de proyectos que no son definidas en el manual de calidad, el monitoreo y evaluación sistemática del proceso de desarrollo y la realización de un proceso sistemático para el aprendizaje de proyectos anteriores.

Los resultados son similares a los reportados en [GAR15], donde todas las prácticas relacionadas con procesos son utilizadas en alguna medida por la industria.

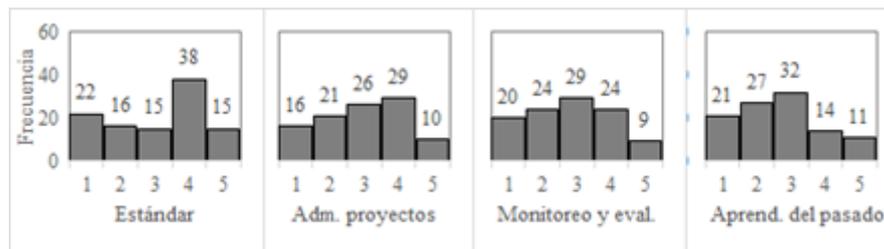


Figura 15 Frecuencia de respuestas para las prácticas relacionadas con procesos

Cuadro 21 Prácticas relacionadas con procesos

Prácticas de procesos	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Tareas basadas en un estándar de procesos del manual de calidad	21	15	14	36	14	3.1	3.5	1.38
Tareas definidas por el equipo de administración de proyectos que no son definidas en el manual de calidad	16	21	25	28	10	3.0	3.0	1.23
Monitoreo y evaluación sistemática del proceso de desarrollo	19	23	27	23	8	2.8	3.0	1.23
Realizan un proceso sistemático para el aprendizaje de proyectos anteriores	20	26	30	13	10	2.7	3.0	1.24

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Desafíos en las prácticas de la ingeniería del software.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación a sus procesos de la SE.

Los desafíos generales que indican los participantes son: cómo incorporar las metodologías ágiles en la administración de proyectos tradicional y en la alta gerencia, cómo mejorar la coordinación en todos los niveles de una organización de desarrollo de software, cómo gestionar el cambio organizacional en organizaciones de tecnología, cómo lograr la optimización de recursos y el incremento en la productividad, cómo poder integrar todos los procesos e implementar procesos de mejora continua, cómo estandarizar los procesos en múltiples equipos, cuáles son las mejores prácticas para mantener la calidad en los modelos de tercerización y cómo administrar los equipos de desarrollo subcontratados, cómo mejorar los aspectos sociales en organizaciones de tecnología, cómo organizar los equipos de desarrollo y aumentar la participación del cliente, cuáles son las mejores prácticas para integrar a los testers en los equipos de desarrollo desde

el inicio del ciclo de vida de desarrollo, cómo transferir el conocimiento de los sistemas al usuario y aumentar la efectividad de las capacitaciones.

### 5.3 Prácticas específicas de la SE

La RQ3 busca determinar tipos de prácticas específicas de la SE para la ingeniería de requerimientos, diseño de *software*, desarrollo, pruebas, mantenimiento, gestión de la configuración, administración de proyectos y gestión de la calidad son usadas por los profesionales.

Para esto se consulta sobre las prácticas comunes que realizan los profesionales para cada una de las áreas de conocimiento del SWEBOK durante el ciclo de vida de desarrollo del *software* y en la administración de sus actividades de *software*.

#### 5.3.1 Requerimientos de *software*

**Prácticas relacionadas con los requerimientos.** En esta pregunta se investiga acerca de las prácticas llevadas a cabo por las organizaciones en la fase de requerimientos. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si el análisis de requerimientos es realizado por un grupo externo y ellos solo reciben la especificación, si los requerimientos se basan solo en el criterio de uno o dos personas de la organización (basado en su idea de innovación), si los requerimientos son documentados claramente o si los detalles de los requerimientos no se encuentran documentados y solo los encargados los conocen (“de memoria”). Se consulta si las liberaciones de versiones son planeadas de antemano, si hay buena comunicación y todos en el equipo se mantienen informados de la estrategia y dirección del producto, si implementan trazabilidad de los requerimientos (*tracking requirements*), si todas las características del producto (*features*) son documentadas y revisadas formalmente, si se realizan prácticas para analizar los requerimientos formalmente que incluye su clasificación, negociación y uso de modelos conceptuales. Finalmente, se consulta si validan los requerimientos mediante inspecciones, revisiones de pares, o *walk throughs*.

La Figura 16 y el Cuadro 22 presentan las frecuencias y porcentajes de las prácticas relacionadas con requerimientos. Aunque no existe una diferencia marcada en la tendencia de las respuestas, las principales prácticas usadas por los participantes son la documentación y revisión formal de todas las características (*features*) del producto, la validación de los requerimientos mediante inspecciones, revisión de pares y los *walk throughs*.

En comparación con los resultados reportados en [GAR15], los valores para estas prácticas son similares, sin embargo con respecto a si implementan trazabilidad de los requerimientos (*tracking requirements*) nuestros resultados son menores. Al igual que en [GAR15], las prácticas menos comunes son que el análisis de requerimientos es realizado por un grupo externo y que los detalles de los requerimientos no se encuentran documentados y solo los encargados los conocen.

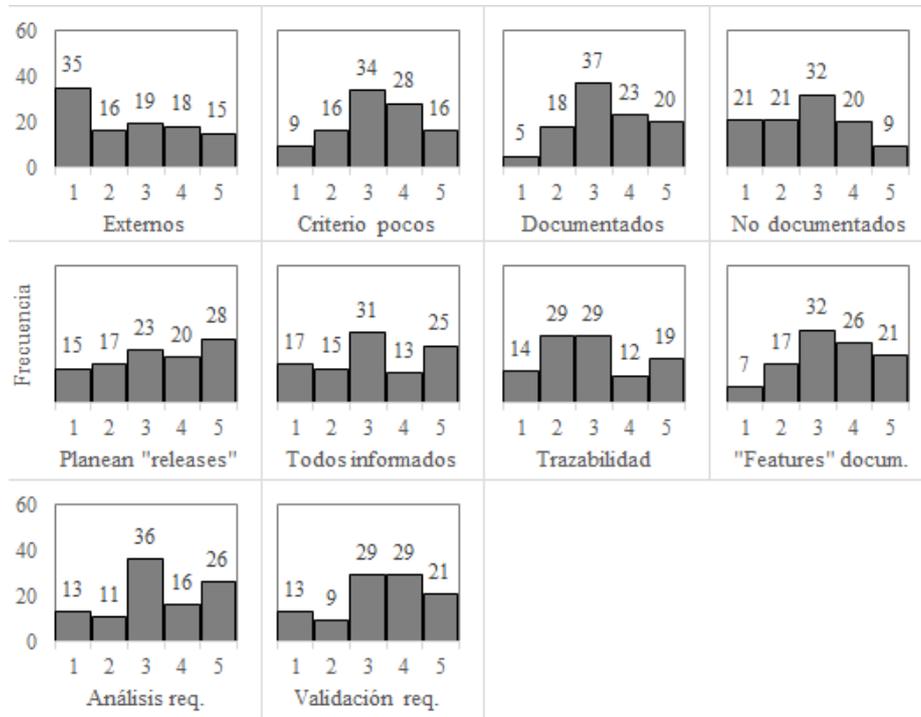


Figura 16 Frecuencia de respuestas para las prácticas relacionadas con requerimientos

Cuadro 22 Prácticas relacionadas con requerimientos

Requerimientos	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Reciben los requerimientos por parte de un grupo externo	34	16	18	17	15	2.6	3	1.47
Los requerimientos se basan en el criterio de uno o dos personas	9	16	33	27	16	3.3	3	1.16
Son los requerimientos documentados claramente	5	17	36	22	19	3.3	3	1.13
Los detalles de los requerimientos no se encuentran documentados y solo los encargados los conocen (“de memoria”)	20	20	31	19	9	2.8	3	1.23
Planean las liberaciones de versiones de antemano	15	17	22	19	27	3.3	3	1.40
Todos se mantienen informados de la estrategia y dirección del producto	17	15	31	13	25	3.1	3	1.39
Implementan trazabilidad de requerimientos ( <i>tracking requirements</i> )	14	28	28	12	18	2.9	3	1.30
Todas las características ( <i>features</i> ) son documentadas y revisadas formalmente	7	17	31	25	20	3.4	3	1.18
Se analizan los requerimientos formalmente (clasificación, negociación, modelos conceptuales)	13	11	35	16	25	3.3	3	1.31
Se validan los requerimientos (inspecciones, revisión de pares, <i>walk throughs</i> )	13	9	29	29	21	3.4	3	1.27

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

Se realiza un análisis estadístico utilizando el coeficiente de correlación de Spearman entre cada par de variables para determinar las relaciones existentes. El coeficiente de correlación de Spearman es una prueba no paramétrica para medir el grado de asociación entre dos variables, que no asume ninguna suposición acerca de la distribución y el tipo de variable.

A partir del análisis de correlación de Spearman entre los pares de variables se determina una relación fuerte, positiva y estadísticamente significativa ( $\alpha = 0.01$ ) entre las siguientes prácticas: (a) se realiza un análisis formal y se validan los requerimientos ( $r_s = .816$ ), (b) todas las características (*features*) son documentadas y revisadas formalmente y los requerimientos son documentados claramente ( $r_s = .611$ ), se realiza un análisis formal ( $r_s = .651$ ) y se validan los requerimientos ( $r_s = .715$ ).

En comparación con los resultados reportados en [GAR15], se confirma la hipótesis de la relación entre el análisis formal y la validación de los requerimientos en las organizaciones. Sin embargo, aunque existe una relación negativa entre requerimientos documentados claramente y los detalles de los requerimientos no se encuentran documentados y solo los encargados los conocen (“de memoria”) ( $r_s = -.087$ ) no se puede confirmar estadísticamente esta hipótesis.

Es necesario un análisis más profundo sobre los factores que influyen en la frecuencia en que son llevadas a cabo las distintas prácticas de requerimientos.

**Notaciones utilizadas en documentos de requerimientos.** La Figura 17 y el Cuadro 23 presentan los tipos de notaciones utilizadas para la especificación de los requerimientos.

El 27% de los participantes indican el uso de casos de uso y escenarios, el 22.8% el uso de "User stories y story boards", el 18.6% el uso de diagramas de actividad UML y un 15.8% el uso del lenguaje natural. Solo el 0.9% de los participantes indican que no se documentan requerimientos o que se utilizan notaciones formales como tablas de decisión o "Software Requirements Engineering Methodology".

En comparación con los resultados reportados en estudios previos, se confirma la utilización predominante de casos de uso y escenarios [GAR15] [AYK09] y "User stories" [GAR15], pero ambos, con porcentajes menores. Sin embargo, el uso de lenguaje natural en nuestro estudio (15.8%) es mucho menor al reportando en [GAR15] (61%) y [AYK09] (69%).

Es necesario un análisis más profundo sobre los factores que influyen en la frecuencia de uso de las distintas notaciones de acuerdo a las metodologías de desarrollo de software, las prácticas de requerimientos y las características de las organizaciones y sus proyectos.

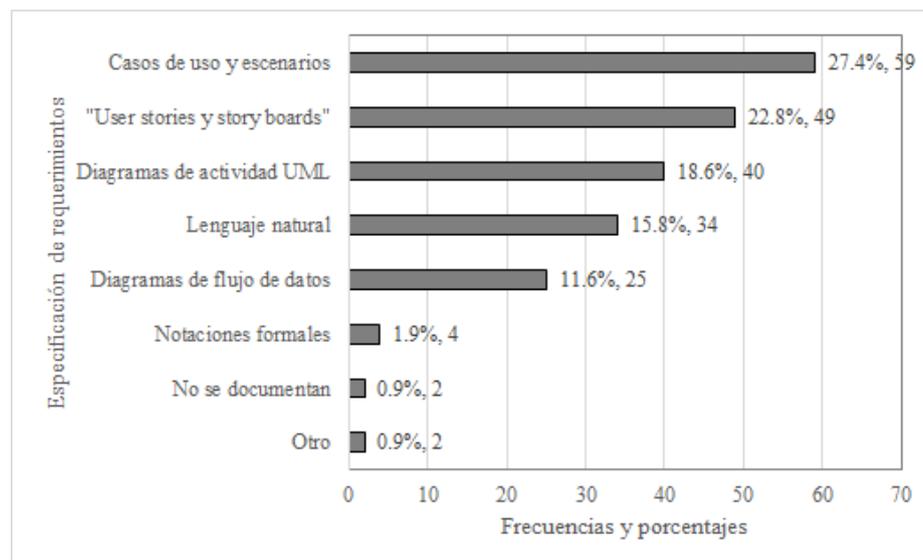


Figura 17 Notaciones para requerimientos

**Cuadro 23 Notaciones para requerimientos**

<b>Notaciones para requerimientos</b>	<b>n</b>	<b>%</b>
Casos de uso y escenarios	59	27.4%
"User stories y story boards"	49	22.8%
Diagramas de actividad UML	40	18.6%
Lenguaje natural	34	15.8%
Diagramas de flujo de datos	25	11.6%
Notaciones de requerimientos formales	4	1.9%
No se documentan requerimientos	2	0.9%
Otro	2	0.9%
<b>Total</b>	<b>215</b>	<b>100.0%</b>

**Desafíos en las prácticas de la ingeniería del software.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación a la ingeniería de requerimientos.

Los desafíos generales que indican los participantes son: cuáles son los mejores métodos de definición de requerimientos, y los métodos efectivos de elicitación de requerimientos, cómo mejorar la especificación y precisión de los requerimientos en la gestión de requerimientos para mejorar su efectividad, cómo mejorar la efectividad de las especificaciones técnicas, cuáles son las mejores prácticas para la evaluación y validación de requerimientos, cómo validar la completitud de los requerimientos, cómo realizar la adopción de modelos formales de requerimientos, cuáles son las mejores prácticas para mejorar la rapidez en la elicitación de los requerimientos sin perder efectividad, cómo lograr la simplificación de documentos de especificación de requerimientos sin perder la efectividad de la especificación y cuáles son los mejores lenguajes comunes entre usuarios y técnicos. Además, cómo determinar los artefactos de requerimientos generan más valor, cuáles son las mejores prácticas para realizar la actualización de especificación de requerimientos a lo largo del ciclo de vida de desarrollo, cómo lidiar con los cambios constantes en los requerimientos en metodologías tradicionales, cómo decidir el mejor tipo de especificación según el tipo de requerimiento y cuál es la mejor manera de escribir los requerimientos, cómo realizar la especificación de los requerimientos de seguridad, cuáles son las mejores prácticas para el realizar el seguimiento de los requerimientos y su trazabilidad, cómo mejorar la productividad de los procesos de análisis de requerimientos y cuál es la documentación mínima esencial para el proceso de desarrollo. Finalmente cómo realizar la representación de requerimientos complejos y cómo hacer entender a los ingenieros nuevos sobre el valor agregado que generan las especificaciones de requerimientos.

### **5.3.2 Diseño de software**

**Prácticas relacionadas con el diseño.** En esta pregunta se investiga acerca de las prácticas llevadas a cabo por las organizaciones en la fase de diseño de *software*. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si se asigna personal dedicado a la arquitectura de software y/o el diseño de alto nivel, si se sigue un estricto proceso de revisión formal del diseño, si el equipo de trabajo realiza revisiones informales del diseño y la frecuencia en la cual se deben tomar decisiones creativas de diseño por tener documentos incompletos producto de la etapa de requerimientos.

La Figura 18 y el Cuadro 24 presentan las frecuencias y porcentajes de las prácticas de diseño. Aunque no existe una diferencia marcada en las tendencias, las principales prácticas usadas son la asignación de personal dedicado a la arquitectura de software y/o el diseño de alto nivel y el equipo de trabajo realiza revisiones informales del diseño.

En comparación con los resultados reportados en [GAR15], los valores para estas prácticas son similares en cuanto a las revisiones informales realizadas por los equipos, sin embargo, con respecto a la frecuencia en la cual se deben tomar decisiones creativas de diseño por tener documentos incompletos producto de la etapa de requerimientos los resultados de nuestro estudio son menores.

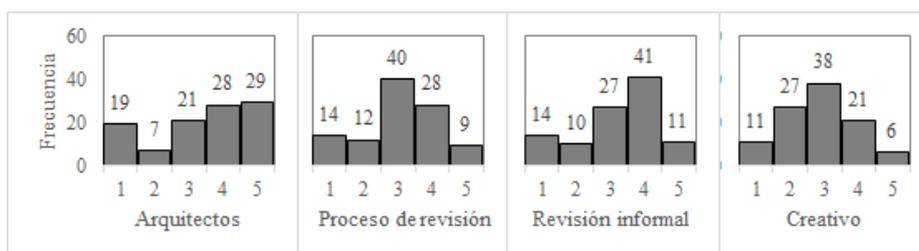


Figura 18 Frecuencia de respuestas para las prácticas de diseño

Cuadro 24 Prácticas de diseño

Prácticas de diseño	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Se asigna personal dedicado a la arquitectura de software y/o diseño de alto nivel	18	7	20	27	28	3.4	4	1.43
Se sigue un estricto proceso de revisión del diseño	14	12	39	27	9	3.1	3	1.14
El equipo realiza revisiones informales del diseño	14	10	26	40	11	3.2	4	1.19
Con frecuencia se deben tomar decisiones de diseño creativas por documentos incompletos de la etapa de requerimientos	11	26	37	20	6	2.8	3	1.05

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

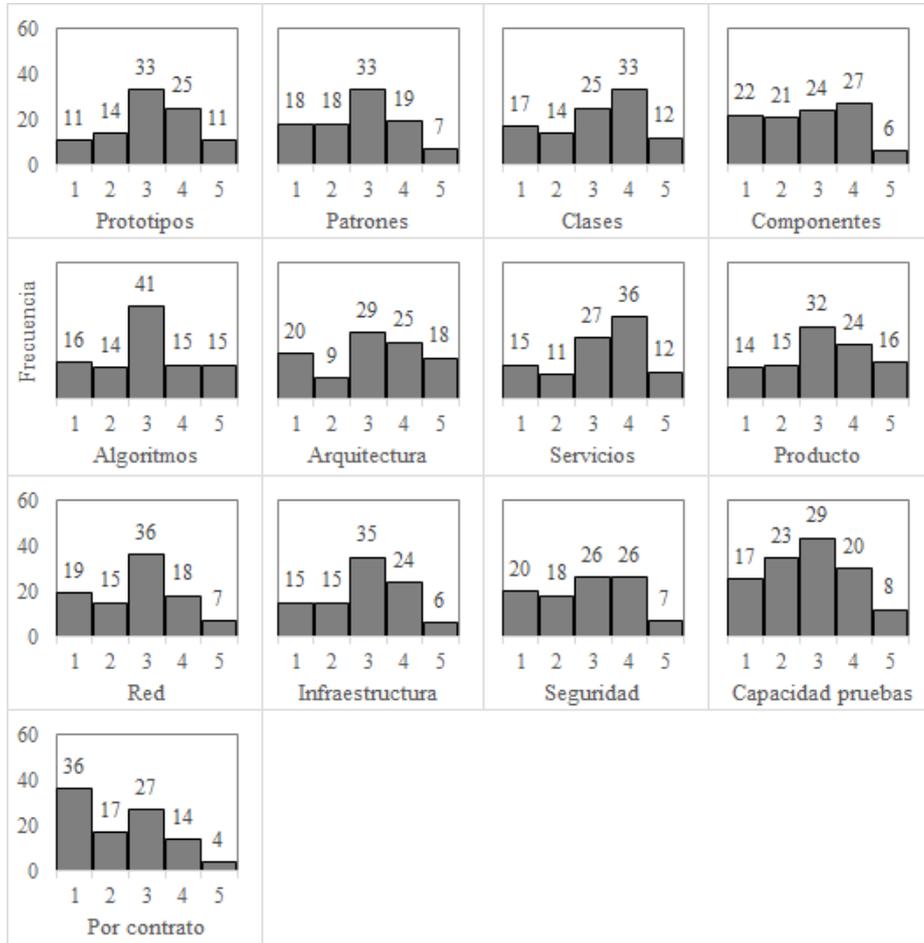
**Tipo de actividades de diseño.** En esta pregunta se investiga acerca de los tipos de actividades realizadas por las organizaciones para el diseño de *software*. La pregunta

se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si diseña por medio de prototipos, si se da el uso sistemático de patrones de diseño, si se realiza el diseño de clases, componentes e interfaz de usuario, si se realiza el diseño y selección de componentes (*component Selection*), si se diseñan los procedimientos algorítmicos y/o computacionales, si se diseña la arquitectura, los servicios, el producto, la red, la infraestructura, la arquitectura de seguridad, el diseño de la capacidad de pruebas (*design for testability*) y finalmente, si los diseños se realizan por contrato.

La Figura 19 y el Cuadro 25 presentan las frecuencias y porcentajes de las actividades de diseño. Aunque no existe una diferencia marcada en las respuestas, las principales tareas realizadas son el diseño de clases, componentes e interfaz de usuario, el diseño por medio de prototipos, y el diseño de los servicios, la arquitectura y el producto.

Al igual que en [GAR15], las actividades de diseño de clases, componentes e interfaz y el diseño de la arquitectura son comunes en la industria y el diseño por contrato es de las prácticas menos populares.



**Figura 19** Frecuencia de respuestas para el tipo de actividades de diseño

**Cuadro 25 Tipo de actividades de diseño**

Actividades de diseño	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Prototipos	12	15	<b>35</b>	<b>27</b>	12	<b>3.12</b>	<b>3</b>	1.16
Patrones de diseño	19	19	<b>35</b>	20	7	2.78	<b>3</b>	1.19
Clases, componentes e interfaz	17	14	25	<b>33</b>	12	<b>3.09</b>	<b>3</b>	<b>1.27</b>
Componentes	22	21	24	<b>27</b>	6	2.74	<b>3</b>	1.24
Procedimientos y algoritmos	16	14	<b>41</b>	15	15	2.99	<b>3</b>	1.24
Arquitectura	20	9	<b>29</b>	25	18	<b>3.12</b>	<b>3</b>	<b>1.36</b>
Servicios	15	11	<b>27</b>	<b>36</b>	12	<b>3.19</b>	<b>3</b>	1.23
Producto	14	15	<b>32</b>	24	16	<b>3.13</b>	<b>3</b>	<b>1.25</b>
Red	20	16	<b>38</b>	19	7	2.78	<b>3</b>	1.19
Infraestructura	16	16	<b>37</b>	25	6	2.91	<b>3</b>	1.14
Arquitectura de seguridad	21	19	<b>27</b>	<b>27</b>	7	2.81	<b>3</b>	1.24
Capacidad de pruebas	18	24	<b>30</b>	21	8	2.78	<b>3</b>	1.20
Por contrato	<b>37</b>	17	<b>28</b>	14	4	2.32	2	1.22

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Atributos de calidad en el diseño.** Esta pregunta investiga acerca de los atributos de calidad en el diseño de *software*. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre la extensibilidad (agregar nuevas capacidades al software sin mayores cambios a la arquitectura), tolerancia a fallos (resiste y se recupera de los fallos), mantenibilidad (facilidad para reparar los errores o agregar nuevas funcionalidades), modularidad (diseño en componentes independientes), confiabilidad (capacidad de ejecutar las funcionalidades requeridas bajo condiciones establecidas por un periodo específico de tiempo), reusabilidad (las piezas de *software* pueden ser reutilizado con una ligera o ninguna modificación en otros proyectos), robustez (capacidad de operar bajo condiciones de estrés y/o tolerar la entrada impredecible de valores no válidos), seguridad (capacidad de soportar actos hostiles e influencias externas) y usabilidad (la interfaz de usuario del software es fácil de usar por el usuario final).

La Figura 20 y el Cuadro 26 presentan las frecuencias y porcentajes con los aspectos de calidad que se consideran importantes en el diseño. Tal como se reporta en [GAR15], los resultados indican que los participantes consideran todos los aspectos importantes en el diseño del *software*. Los atributos de seguridad y usabilidad son los atributos reportados como los más importantes, contrario a [GAR15], sin embargo los valores de las tendencias para todos los atributos de calidad son similares entre los estudios comparados.

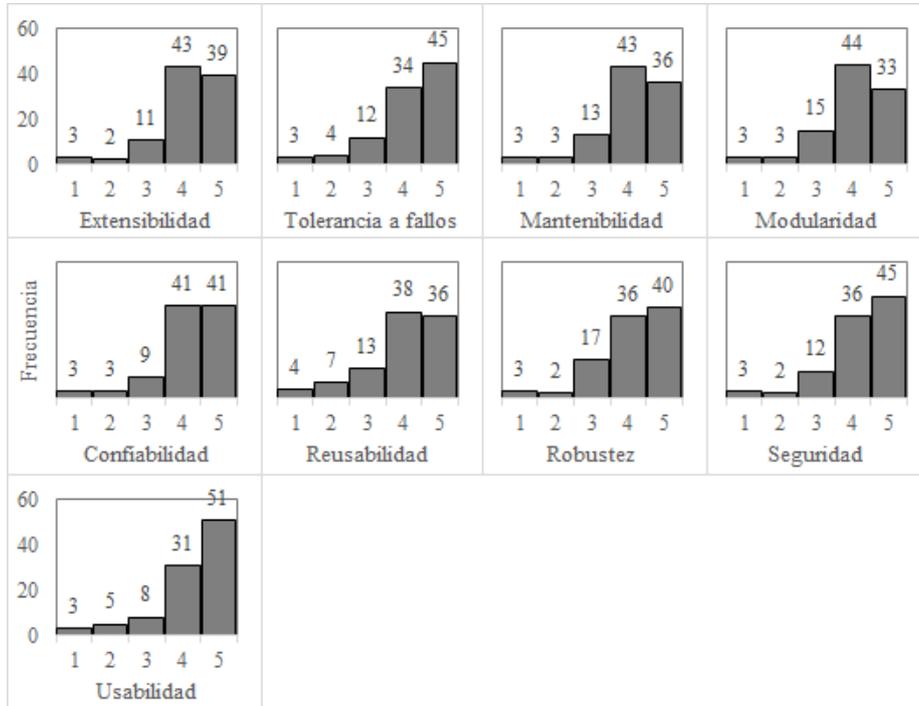


Figura 20 Frecuencia de respuestas para los atributos de calidad en el diseño

Cuadro 26 Atributos de calidad en el diseño

Atributos de calidad	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Extensibilidad	3	2	11	<b>44</b>	<b>40</b>	4.15	4	0.92
Tolerancia a fallos	3	4	12	35	<b>46</b>	4.16	4	1.00
Mantenibilidad	3	3	13	<b>44</b>	37	4.08	4	0.95
Modularidad	3	3	15	<b>45</b>	34	4.03	4	0.95
Confiabilidad	3	3	9	<b>42</b>	<b>42</b>	4.18	4	0.95
Reusabilidad	4	7	13	<b>39</b>	37	3.97	4	<b>1.08</b>
Robustez	3	2	17	37	<b>41</b>	4.10	4	0.97
Seguridad	3	2	12	37	<b>46</b>	<b>4.20</b>	4	0.95
Usabilidad	3	5	8	32	<b>52</b>	<b>4.24</b>	<b>5</b>	<b>1.02</b>

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Desafíos en las prácticas de la ingeniería del software.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación a las prácticas de diseño.

Los desafíos generales que indican los participantes son: cuáles son las mejores prácticas para diseñar sistemas fáciles de entender, cómo diseñar las interfaces de usuario aplicando las teorías de usabilidad y experiencia del usuario. Además, cómo mejorar

en la productividad en el diseño de modelos conceptuales, cuál es el nivel de detalle esencial para un diseño completo, cuáles son las tareas de diseño que generan más valor y cuáles son los artefactos esenciales necesarios. Cómo decidir la arquitectura adecuada según la solución requerida, cómo realizar un diseño robusto, flexible, escalable y re-utilizable, cuál es la manera sistemática de aplicar los patrones de diseño, cuáles son las mejores prácticas para adoptar estándares en la práctica profesional, cómo diseñar software para que tenga un buen rendimiento, cómo se diseñan aspectos de seguridad, cómo unificar la documentación y mantener la trazabilidad entre requerimientos y diseño, cómo se realizan los procesos de validación del diseño y validación contra los requerimientos, cómo se realiza el diseño en equipos de trabajo colaborativos, cómo mantener el diseño actualizado a lo largo del ciclo de vida de desarrollo, cómo se evalúa la calidad del diseño, cuáles son las mejores prácticas para arquitecturas *multi-tenant* y cómo se debe realizar la capacitación en diseño de software orientado a objetos para que sea más efectiva.

### 5.3.3 Desarrollo o construcción de *software* (codificación)

**Prácticas relacionadas con el desarrollo.** En esta pregunta se investiga acerca de las prácticas llevadas a cabo por las organizaciones en la fase de codificación. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre prácticas como la programación en pares ("*pair programming*"), actividades de refactorización ("*refactoring*"), actividades de inspección de código y revisiones por pares ("*code inspections and peer reviews*"), análisis estático de código fuente usando herramientas automatizadas, generación automática de documentación usando herramientas como *JavaDocs* ("*systematic built-in code documentation*") y medición, administración y minimización de la complejidad del código. Además se consulta sobre si los desarrolladores son responsables por el diseño de alto nivel y la implementación, si el diseño y la codificación son realizados en conjunto, si se realizan los *builts* diariamente (al menos una vez por día) o si se realizan los "*builts*" cuando una característica o funcionalidad se completa. Finalmente, se consulta si el proceso de los "*builts*" está documentado y se le da seguimiento, si se cuenta con una persona encargada de los "*builts*" y si el código fuente es revisado antes de realizar un "*built*".

La Figura 21 y el Cuadro 27 presentan las frecuencias y porcentajes de las prácticas relacionadas con el desarrollo. Las prácticas reportadas como más frecuentes son que los desarrolladores son responsables por el diseño de alto nivel y la implementación, el diseño y la codificación son realizados en conjunto, y el proceso de los "*builts*" está documentado y se le da seguimiento. Estas prácticas coinciden con las reportadas en [GAR15] con valores similares. En el caso de la práctica de realizar los *builts* cuando una característica o funcionalidad se completa la tendencia es similar entre ambos estudios. Las prácticas menos frecuentes son las de la generación automática de documentación usando herramientas como *JavaDocs* ("*systematic built-in code documentation*") y la medición, administración y minimización de la complejidad del código.

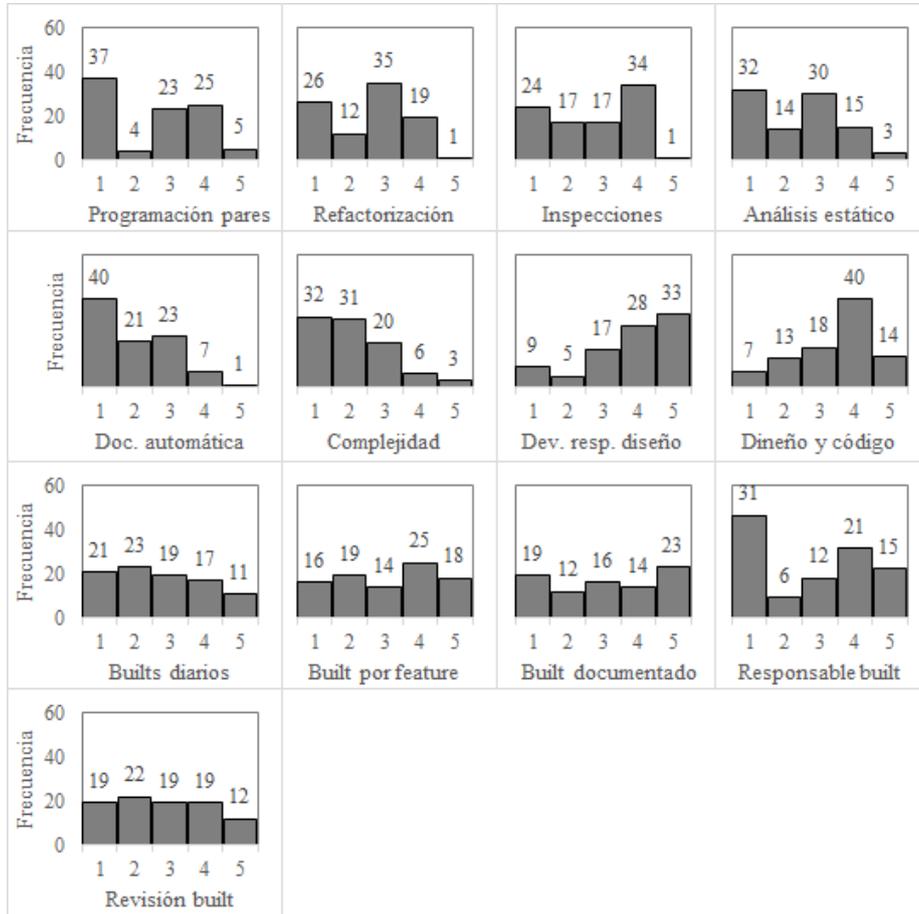


Figura 21 Frecuencia de respuestas para las prácticas relacionadas con el desarrollo

Cuadro 27 Prácticas relacionadas con el desarrollo

Prácticas relacionadas con el desarrollo	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Programación en pares (" <i>pair programming</i> ")	<b>39</b>	4	24	<b>27</b>	5	2.54	3	1.38
Refactorización (" <i>refactoring</i> ")	<b>28</b>	13	<b>38</b>	20	1	2.54	3	1.14
Inspección de código y revisiones por pares (" <i>code inspections and peer reviews</i> ")	26	18	18	<b>37</b>	1	2.69	3	1.24
Análisis estático de código fuente con herramientas automatizadas	<b>34</b>	15	<b>32</b>	16	3	2.39	3	1.20
Generación automática de documentación usando herramientas como JavaDocs (" <i>systematic built-in code documentation</i> ")	<b>43</b>	23	25	8	1	2.00	2	1.05
Medición y minimización de complejidad del código	<b>35</b>	<b>34</b>	22	7	3	2.10	2	1.06
Los desarrolladores son responsables por el diseño de alto nivel y la implementación	10	5	18	<b>30</b>	<b>36</b>	<b>3.77</b>	<b>4</b>	1.27
El diseño y la codificación son realizados en conjunto	8	14	20	<b>43</b>	15	<b>3.45</b>	<b>4</b>	1.14
Se realizan los <i>builts</i> diariamente (al menos una vez por día)	23	25	21	19	12	2.71	3	1.34
Se realizan los <i>builts</i> cuando una característica o funcionalidad se completa	17	21	15	<b>27</b>	20	3.11	3	<b>1.40</b>
El proceso de los " <i>builts</i> " está documentado y se le da seguimiento	23	14	19	17	<b>27</b>	<b>3.12</b>	3	<b>1.52</b>
Contamos con una persona encargada de los " <i>builts</i> "	<b>36</b>	7	14	25	18	2.80	3	<b>1.57</b>
El código fuente es revisado antes de realizar un " <i>built</i> "	21	24	21	21	13	2.81	3	1.34

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

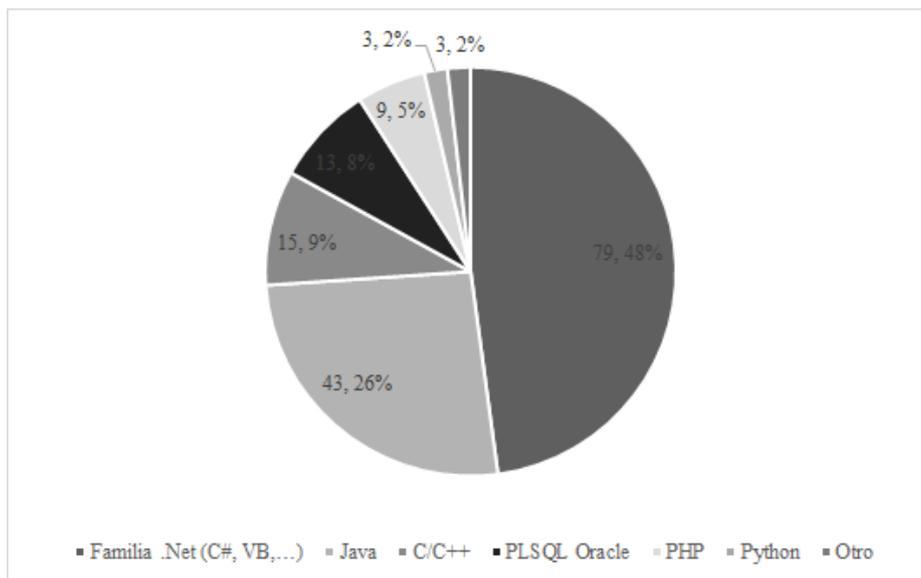
Se realiza un análisis de las prácticas relacionadas con la fase de desarrollo utilizando el coeficiente de correlación de Spearman entre cada par de variables. A partir del análisis de correlación de Spearman se determina una relación fuerte, positiva y estadísticamente significativa ( $\alpha = 0.01$ ) para la hipótesis planteada en [GAR15] entre la programación en pares ("*pair programming*") y la inspección de código y revisiones por pares ("*code inspections and peer reviews*") ( $r_s = .734$ ). Además se determinan relaciones fuertes, positivas y estadísticamente significativas ( $\alpha = 0.01$ ) entre las siguientes prácticas: (a) prácticas de refactorización ("*refactoring*") y programación en pares ( $r_s = .726$ ) e inspección de código y revisiones por pares ( $r_s = .706$ ), (b) se realizan

los *builds* diariamente cuando se ha generado una característica o funcionalidad completa ( $r_s = .704$ ), con el proceso documentado ( $r_s = .722$ ) y personal asignado ( $r_s = .673$ ).

Es necesario un análisis más profundo sobre los factores que influyen en la frecuencia en que son llevadas a cabo las distintas prácticas de desarrollo.

**Lenguajes de programación.** En esta pregunta se consulta acerca de los lenguajes de programación más utilizados en desarrollo. Los participantes pueden reportar todos los lenguajes que utilizan.

La Figura 22 y el Cuadro 28 presentan las frecuencias y porcentajes del uso de los distintos lenguajes de programación. El 46% de los participantes reportan el uso de más de un lenguaje de programación. Los lenguajes más utilizados son los de la familia .Net (C#, VB,...) con un 48%, seguido de Java con un 26%. C/C++ tienen un porcentaje del 9%, PL/SQL Oracle un 7.9% y PHP con un 5.5%. En comparación con otros estudios que reportaron el uso de lenguajes de programación [GAR13] [GAR15], podemos observar que Java y .Net siguen manteniendo un alto nivel de utilización en los países estudiados.



**Figura 22 Lenguajes de programación**

**Cuadro 28 Lenguajes de programación**

<b>Lenguajes</b>	<b>n</b>	<b>%</b>
Familia .Net (C#, VB, ...)	79	47.9%
Java	43	26.1%
C/C++	15	9.1%
PLSQL Oracle	13	7.9%
PHP	9	5.5%
Python	3	1.8%
Otro	3	1.8%
<b>Total</b>	<b>165</b>	<b>100.0%</b>

**Desafíos en las prácticas de la ingeniería del software.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación con el desarrollo de software.

Los desafíos generales que indican los participantes son: cuáles son las mejores prácticas para adoptar el uso de metodologías de desarrollo cero defectos, cómo lograr un buen nivel de automatización en todos los niveles del proceso de desarrollo, cómo desarrollar y gestionar componentes reutilizables, cómo gestionar la reutilización de código, cómo codificar para lograr la escalabilidad de las aplicaciones, cuáles son las mejores prácticas para la codificación y documentación del código fuente, cómo codificar de acuerdo a las mejores prácticas de seguridad, cómo lograr un buen nivel de *clean code* y cuál es la mejor forma de realizar los procesos de *refactoring*, cómo garantizar la calidad del código, como mejorar el acoplamiento y cohesión, cuáles son las métricas clave la medición de la calidad del código fuente, y cuáles son las mejoras prácticas para la estandarización en la programación de los desarrolladores, finalmente, cómo enseñar los principios básicos de programación tal como *SOLID* con mayor efectividad.

### 5.3.4 Pruebas de *software*

**Prácticas relacionadas con pruebas.** En esta pregunta se investiga acerca de las prácticas llevadas a cabo por las organizaciones en la fase de pruebas de *software*. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre prácticas como las pruebas de unidad ("*unit tests for code*"), si las pruebas de unidad son revisadas formalmente, si se tiene un equipo separado para realizar las pruebas del producto, si los desarrolladores y testers trabajan en estrecha colaboración y si los desarrolladores prueban el producto antes de una liberación del producto. Además, se consulta si un gerente, experto en el dominio del negocio (cliente o apoyo) ayuda a probar el producto y si todas las nuevas características ("*features*") son probadas independientemente por un equipo de pruebas.

La Figura 23 y el Cuadro 29 presentan las frecuencias y porcentajes de cada una de las prácticas relacionadas con pruebas. Las prácticas reportadas como más frecuentes son que un gerente, experto en el dominio del negocio (cliente o apoyo) ayuda a probar el producto, los desarrolladores prueban el producto antes de cada liberación y el uso

de las pruebas de unidad ("*unit tests for code*"). Veintiún participantes indican que siempre se tiene un equipo separado para realizar las pruebas del producto y 23 que casi siempre.

Las prácticas coinciden con las reportadas en [GAR15] donde los valores son similares con excepción en la que un gerente, experto en el dominio del negocio (cliente o apoyo) ayuda a probar el producto.

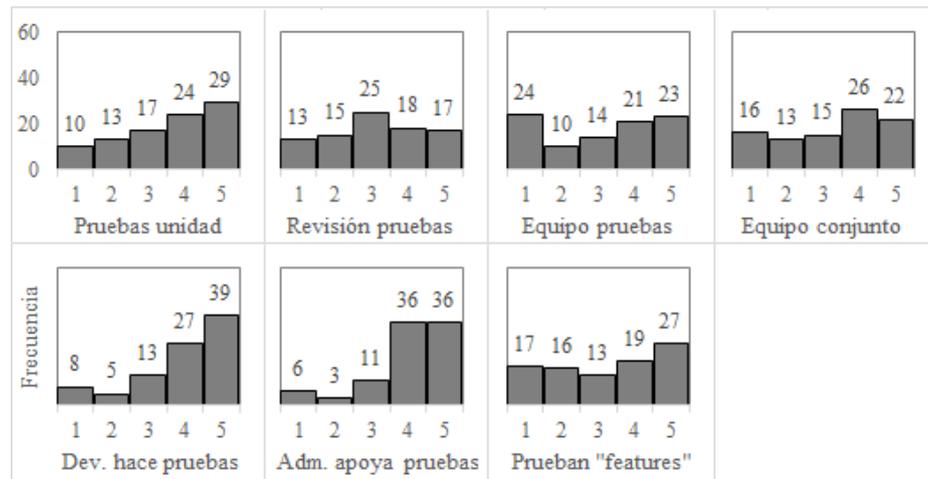


Figura 23 Frecuencia de respuestas para las prácticas relacionadas con pruebas

Cuadro 29 Prácticas relacionadas con pruebas

Prácticas relacionadas con pruebas	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Pruebas de unidad (" <i>unit tests</i> ")	11	14	18	26	<b>31</b>	3.53	<b>4</b>	1.35
Las pruebas de unidad son revisadas formalmente	15	17	<b>28</b>	20	19	3.13	3	1.32
Se tiene un equipo separado para realizar las pruebas del producto	26	11	15	23	25	3.10	3	<b>1.55</b>
Los desarrolladores y testers trabajan en estrecha colaboración	17	14	16	<b>28</b>	24	3.27	<b>4</b>	1.42
Los desarrolladores prueban el producto antes de la liberación	9	5	14	<b>29</b>	<b>42</b>	3.91	<b>4</b>	1.25
Un gerente, experto en el dominio del negocio (cliente o apoyo) ayuda a probar el producto	7	3	12	<b>39</b>	<b>39</b>	<b>4.01</b>	<b>4</b>	1.11
Las nuevas características (" <i>features</i> ") son probadas independientemente por un equipo de pruebas	18	17	14	21	<b>29</b>	3.25	3.5	1.50

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Fases de pruebas en el ciclo de vida de desarrollo.** En esta pregunta se investiga acerca de las fases de pruebas que se realizan en el ciclo de vida de desarrollo. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si realizan las pruebas de acuerdo a "*Test driven Development*" (TDD) (también conocido como "*Test first Development*") o "*Test last Development*" (TLD) donde las pruebas se realizan una vez que el *software* está totalmente desarrollado.

La Figura 24 y el Cuadro 30 presentan las frecuencias y porcentajes para cada una de las fases estudiadas. La mayoría de los participantes reportan que realizan la fase de pruebas, de manera tradicional, una vez que el *software* está totalmente desarrollado ("*Test last Development*"). Esta práctica coincide con la reportada en [GAR13] [GAR15] donde la tendencia es la misma.

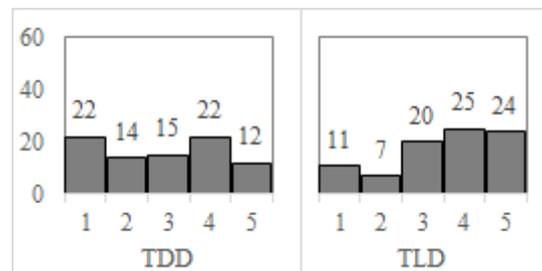


Figura 24 Frecuencia de respuestas para la fase de pruebas

Cuadro 30 Fase de pruebas

Fase de pruebas	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
" <i>Test-driven Development</i> " (TDD)	26	16	18	26	14	2.86	3	1.42
" <i>Test-last Development</i> "(TLD)	13	8	23	29	28	3.51	4	1.32

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Tipos y niveles de pruebas.** En esta pregunta se investiga acerca de los tipos y niveles de pruebas que se realizan en la industria. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si realizan pruebas de unidad, pruebas de integración, pruebas de aceptación de usuario, pruebas funcionales, pruebas de sistema, pruebas de rendimiento, pruebas de carga o estrés, pruebas de seguridad y/o pruebas de usabilidad.

La Figura 25 y el Cuadro 31 presentan las frecuencias y porcentajes para cada uno de los tipos y niveles de pruebas. La mayoría de los participantes reportan que realizan pruebas de aceptación de usuario, pruebas funcionales, pruebas de sistema y pruebas de integración. Estas pruebas coinciden con las reportadas en [GAR13] [GAR15] donde la tendencia se mantiene. La prueba menos común es la prueba de carga/estrés de igual manera que lo reportado en [GAR15].

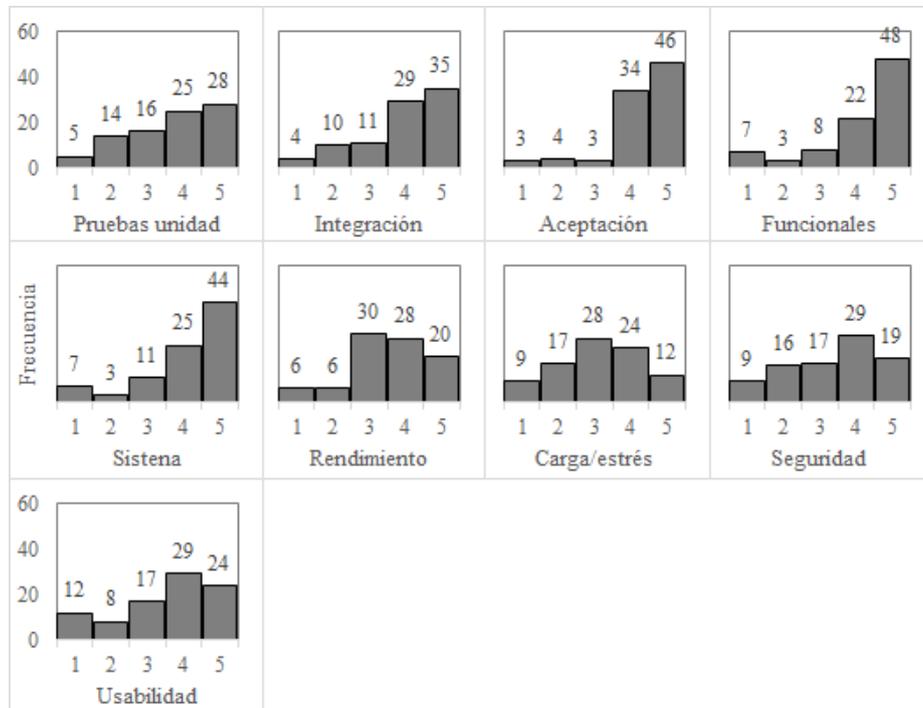


Figura 25 Frecuencia de respuestas para los tipos y niveles de pruebas

Cuadro 31 Tipos y niveles de pruebas

Tipos y niveles de pruebas	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Pruebas de unidad	6	16	18	28	<b>32</b>	3.65	4	1.24
Pruebas de integración	4	11	12	<b>33</b>	<b>39</b>	3.91	4	1.17
Pruebas de aceptación de usuario	3	4	3	<b>38</b>	<b>51</b>	<b>4.29</b>	<b>5</b>	0.97
Pruebas funcionales	8	3	9	25	<b>55</b>	<b>4.15</b>	<b>5</b>	1.22
Pruebas de sistema	8	3	12	28	<b>49</b>	4.07	4	1.21
Pruebas de rendimiento	7	7	<b>33</b>	<b>31</b>	22	3.56	4	1.11
Pruebas de carga/estrés	10	19	<b>31</b>	27	13	3.14	3	1.18
Pruebas de seguridad	10	18	19	<b>32</b>	21	3.37	4	<b>1.28</b>
Pruebas de usabilidad	13	9	19	<b>32</b>	27	3.50	4	<b>1.33</b>

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Técnicas de diseño de casos de prueba.** En esta pregunta se investiga sobre las técnicas de diseño de casos de prueba. Los participantes son consultados sobre si utilizan o no una técnica específica para generar los casos de prueba, si se apoyan en técnicas basadas en modelos (ej.: UML), si utilizan el análisis de valores límite ("*boundary value analysis*"), técnicas de análisis de código o caja blanca ("*source code analysis, white-*

box testing") y/o clases de equivalencia ("equivalence classing o category partitioning").

La Figura 26 y el Cuadro 32 presentan las frecuencias y porcentajes para cada una de las técnicas. Las técnicas basadas en modelos son las más utilizadas (22%), seguido del análisis de valores límite (14.4%), caja blanca (14.4%), clases de equivalencia (9.8%). Aunque clases de equivalencia y valores límite son técnicas de caja negra, los participantes indicaron en la opción otros, la de caja negra (5.3%). Ninguno menciona la técnica de caja negra: análisis de cause y efecto. Al igual que en [GAR15], la no utilización de una técnica específica es alta (34.1%). Es necesario más análisis sobre los factores que influyen en el uso de las técnicas de diseño de pruebas de acuerdo a las características de los participantes.

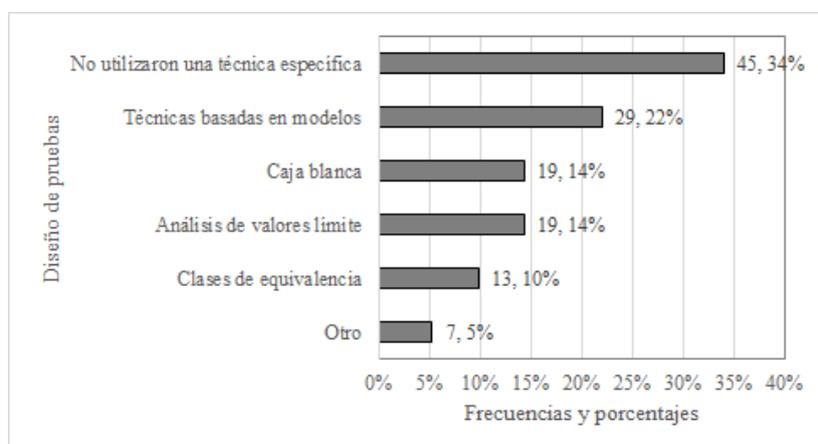


Figura 26 Frecuencia de respuestas para las técnicas de diseño de casos de prueba

Cuadro 32 Técnicas de diseño de casos de prueba

Técnica de diseño	n	%
No utilizaron una técnica específica para generar los casos de prueba	45	34.1%
Técnicas basadas en modelos (ej.: UML)	29	22.0%
Análisis de valores límite ("boundary value analysis")	19	14.4%
Caja blanca ("source code analysis, white-box testing")	19	14.4%
Clases de equivalencia ("equivalence classing category partitioning")	13	9.8%
Otros ("black box")	7	5.3%
<b>Total</b>	<b>132</b>	<b>100.0%</b>

**Automatización de pruebas.** En esta pregunta se comparan las pruebas manuales versus las pruebas automatizadas. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

La Figura 27 y el Cuadro 33 presentan las frecuencias y porcentajes del uso de pruebas automatizadas. En nuestro estudio prevalecen las pruebas manuales versus las automatizadas, sin embargo se observa que la adopción de pruebas automatizadas donde 31 participantes indican que es un enfoque que han adoptado. En comparación con lo reportado por [GAR13] [GAR15] la automatización de las pruebas es similar. Es necesario más análisis sobre los factores que influyen en la automatización de pruebas.

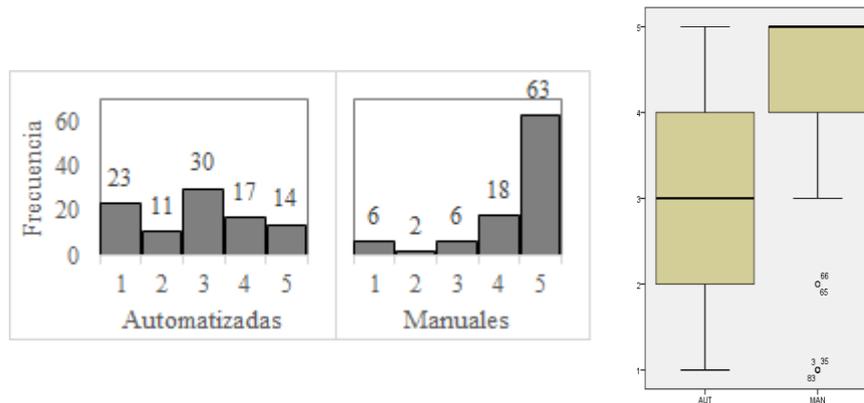


Figura 27 Frecuencia de respuestas para la automatización de pruebas

Cuadro 33 Automatización de pruebas

Automatización	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Pruebas automatizadas	24	12	32	18	15	2.87	3	1.36
Pruebas manuales	6	2	6	19	66	4.37	5	1.12

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Métricas de cobertura de código.** En esta pregunta se consulta sobre el uso de las métricas de cobertura de código o cobertura de pruebas. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre la utilización de las métricas "Line (statement) coverage", "Decision (branch) coverage", "Condition coverage" y/o "MC/DC coverage". La Figura 28 y el Cuadro 34 presentan las frecuencias y porcentajes del uso de las métricas de cobertura. Al igual que lo reportado en [GAR15] la utilización de métricas de cobertura es limitada por lo que es necesario que los profesionales conozcan más de los beneficios de estas mediciones. La métrica más utilizada en nuestro estudio es "Line (statement) coverage". Este estudio solo se contempla criterios de "control-flow" y no de "data-flow" lo que debe tomarse en cuenta en una futura replicación.

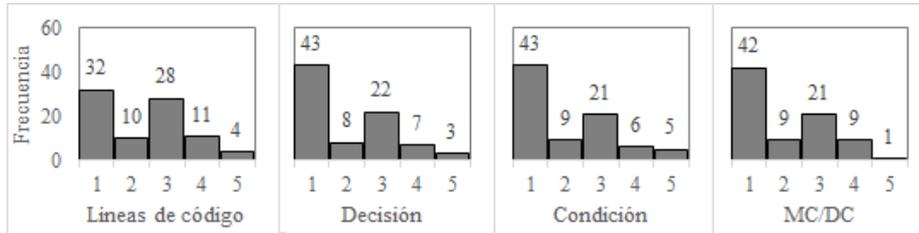


Figura 28 Frecuencia de respuestas para las métricas de cobertura

Cuadro 34 Métricas de cobertura

Métricas de cobertura	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
"Line (statement) coverage"	38	12	33	13	5	2.35	3	1.24
"Decision (branch) coverage"	52	10	27	8	4	2.02	1	1.21
"Condition coverage"	51	11	25	7	6	2.06	1	1.26
"MC/DC coverage"	51	11	26	11	1	2.00	1	1.15

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Métricas de calidad.** En esta pregunta se consulta sobre el uso de las métricas de calidad. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre la utilización de las métricas de defectos por línea de código, la productividad de detección de errores, los defectos detectados por un periodo de tiempo, los casos de prueba ejecutados en un periodo de tiempo y/o los casos de prueba de aceptación aprobados.

La Figura 29 y el Cuadro 35 presentan las frecuencias y porcentajes del uso de las métricas de calidad. Al igual que lo reportado en [GAR15] la utilización de estas métricas es limitada por lo que es necesario que los profesionales conozcan más de los beneficios de estas mediciones. Las métricas de calidad más frecuentes en nuestro estudio son los casos de pruebas de aceptación aprobados y los defectos detectados por un periodo de tiempo.

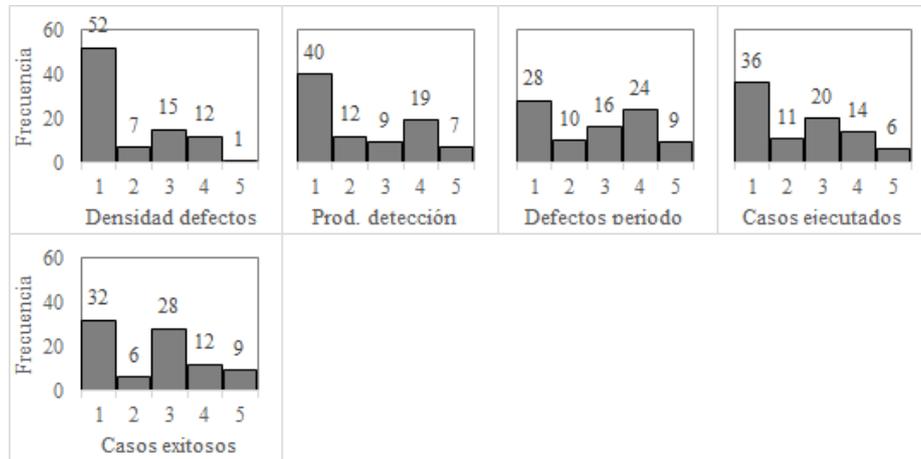


Figura 29 Frecuencia de respuestas para las métricas de calidad

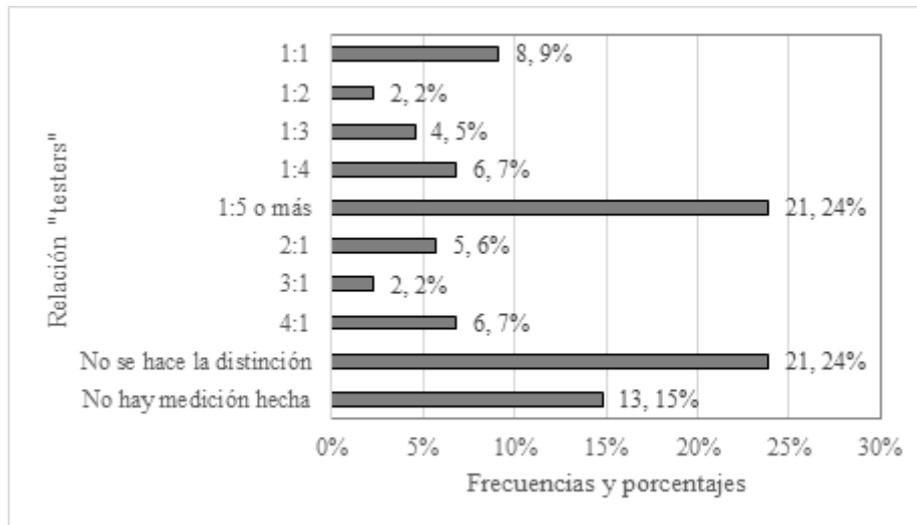
Cuadro 35 Métricas de calidad

Métricas relacionadas con pruebas	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Defectos por línea de código	60	8	17	14	1	1.89	1	1.20
Productividad de detección de errores	46	14	10	22	8	2.32	2	1.44
Defectos detectados en un periodo de tiempo	32	11	18	28	10	2.72	3	1.43
Casos de prueba ejecutados en un periodo de tiempo	41	13	23	16	7	2.34	2	1.35
Casos de prueba de aceptación aprobados	37	7	32	14	10	2.54	3	1.38

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Relación entre testers y desarrolladores.** La relación entre el número de testers:desarrolladores en las organizaciones ha sido un punto de discusión en las organizaciones de *software*. En esta pregunta se consulta sobre esta relación en las organizaciones estudiadas.

En la Figura 30 y el Cuadro 36 podemos observar que no existe una tendencia predominante entre todas las combinaciones. La tendencia principal es la de un tester por cada 5 o más desarrolladores (24%). Gran cantidad de participantes indican que las organizaciones no hacen una distinción entre testers y desarrolladores o no realizan ninguna medición al respecto. Al igual que lo reportado en [GAR15] la relación 1:5 es la más popular en las organizaciones. Aunque no existe una combinación optima de testers:desarrolladores para todas las organizaciones, si se requiere que estas puedan medir y calibrar la relación de sus equipos de trabajo para mejorar su productividad y calidad. En algunos casos contar con pocos testers podría retrasar las liberaciones del producto e influir en la calidad del producto.



**Figura 30** Relación entre *testers* y desarrolladores

**Cuadro 36** Relación entre *testers* y desarrolladores

Relación	n	%
1:1	8	9%
1:2	2	2%
1:3	4	5%
1:4	6	7%
1:5 o más	21	24%
2:1	5	6%
3:1	2	2%
4:1	6	7%
No se hace la distinción	21	24%
No hay medición hecha	13	15%
<b>Total</b>	<b>88</b>	<b>100%</b>

**Criterios de finalización de pruebas.** Esta pregunta consulta acerca de cuál es el momento en que se deben detener las pruebas y liberar el producto. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados si las pruebas se deben detener cuando se alcanza un umbral de cobertura de código (%), si se realiza por un tiempo de duración fijo, o por un presupuesto fijo, si la decisión se realiza de manera informal o si debe detenerse cuando no se encuentran más errores y/o cuando todos los casos de prueba se han ejecutado hasta no encontrar más errores.

Al igual que en [GAR15], la Figura 31 y el Cuadro 37 muestran que el criterio más utilizado para la finalización de pruebas es el de ejecutar todos los casos de prueba hasta no encontrar más defectos. De mismo modo, la adopción de métricas de cobertura se reporta como de uso limitado en nuestra industria.

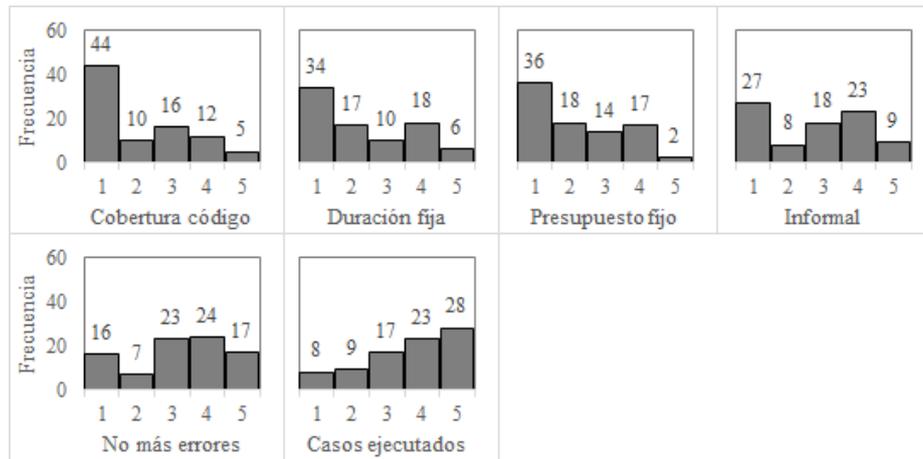


Figura 31 Frecuencia de respuestas para los criterios de finalización de pruebas

Cuadro 37 Criterios de finalización de pruebas

Criterios finalización	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Cobertura de código (%)	51	11	18	14	6	2.13	1	1.33
Tiempo de duración fijo	40	20	12	21	7	2.35	2	1.38
Presupuesto fijo	41	21	16	20	2	2.21	2	1.24
Informal	32	9	21	27	11	2.75	3	1.42
No más errores	18	8	26	28	20	3.22	3	1.36
Todos los casos de prueba hasta no encontrar más defectos	9	11	20	27	33	3.64	4	1.30

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Desafíos en las prácticas de la ingeniería del software.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación con las pruebas de software.

Los desafíos generales que indican los participantes son: cómo garantizar la calidad de los productos y la calidad del código fuente, cuáles son las mejores prácticas para automatizar las pruebas, pruebas de unidad, pruebas de aceptación, pruebas integración y de sistema, automatización cuándo se utilizan metodologías TDD, mejores prácticas para diseñar las pruebas unitarias y pruebas de integración, cómo utilizar las métricas de cobertura de código en los procesos de pruebas, cómo lidiar con la baja *testability* del software, cómo garantizar la calidad de las pruebas, cuáles son las mejores prácticas para la corrección del código y cómo mejorar la productividad en las pruebas, cuáles

son las mejores prácticas para realizar pruebas por iteraciones, cuál es la documentación esencial para las pruebas y cuáles son las mejoras prácticas en la gestión y configuración de los distintos ambientes de pruebas (integración, pruebas, preproducción, entre otros).

### 5.3.5 Mantenimiento de *software*

**Tipo de actividades de mantenimiento.** En esta pregunta se investiga acerca de las prácticas llevadas a cabo por las organizaciones en el mantenimiento del *software*. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre las actividades realizadas en mantenimiento correctivo (modificación reactiva del producto de software que se realiza después de la entrega para corregir problemas descubiertos), mantenimiento adaptativo (modificación del producto de software que se realiza después de la entrega para mantener un producto utilizable en un entorno que ha cambiado), y/o mantenimiento perfectivo (modificación del producto de software después de la entrega para mejorar el rendimiento o mantenibilidad).

La Figura 32 y el Cuadro 38 presentan las frecuencias y porcentajes de las actividades de mantenimiento. El mantenimiento correctivo es la actividad más utilizada y el mantenimiento perfectivo la menos usada. Estas prácticas coinciden con las reportadas en [GAR15]. Los valores reportados para nuestro estudio son levemente mayores tanto para el mantenimiento correctivo como el perfectivo.

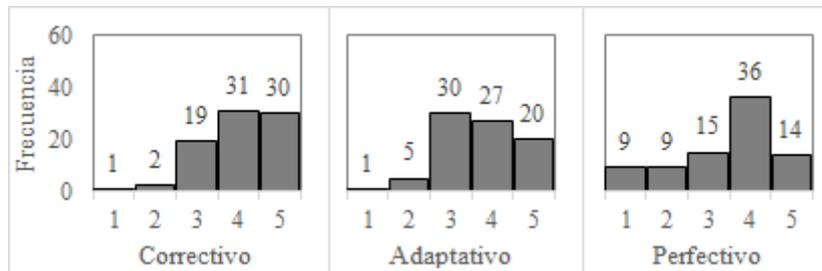


Figura 32 Frecuencia de respuestas para el tipo de actividades de mantenimiento

Cuadro 38 Tipo de actividades de mantenimiento

Mantenimiento	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Correctivo	1	2	23	37	36	4.05	4	0.90
Adaptativo	1	6	36	33	24	3.72	4	0.94
Perfectivo	11	11	18	43	17	3.45	4	1.21

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

Al consultar por las actividades que representan una mayor nivel de desafío en mantenimiento, los participantes consideran que el mantenimiento perfectivo y adaptativo

son los más difíciles de realizar, que son las que se realizan con menos frecuencia, tal como se muestra en la Figura 33 y el Cuadro 39. La misma tendencia se presenta en [GAR15].

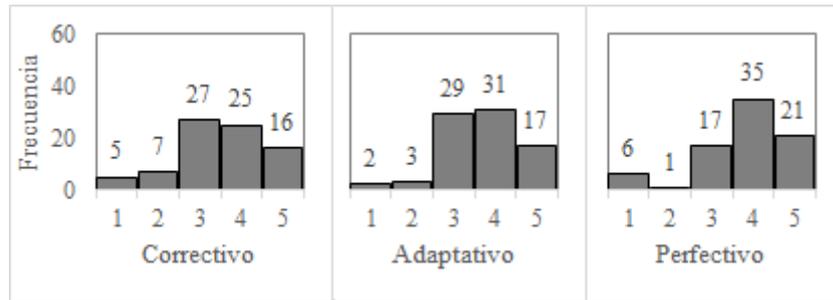


Figura 33 Frecuencia de respuestas para el nivel de desafío en mantenimiento

Cuadro 39 Nivel de desafío en mantenimiento

Desafíos	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Correctivo	6	9	34	31	20	3.50	4	1.10
Adaptativo	2	4	35	38	21	3.71	4	0.92
Perfectivo	8	1	21	44	26	3.80	4	1.08

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

Al indagar sobre algunos de tipos de desafíos que se presentan en las labores de mantenimiento, se consulta sobre las diferentes situaciones que pueden dificultar estas labores. La Figura 34 y el Cuadro 40 muestran que los principales desafíos que enfrentan los profesionales en labores de mantenimiento son los diferentes estilos de programación usados por los desarrolladores, la documentación insuficiente, y en menor medida la falta de herramientas de soporte para el mantenimiento. Los resultados coinciden con la tendencia reportada por [GAR15].

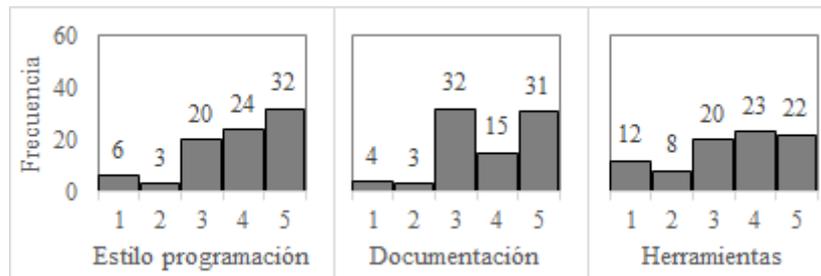


Figura 34 Frecuencia de respuestas para los desafíos en mantenimiento

**Cuadro 40 Desafíos en mantenimiento**

Desafíos en mantenimiento	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Diferentes estilos de programación usados por los desarrolladores originales	7	4	24	28	<b>38</b>	<b>3.86</b>	<b>4</b>	1.18
Documentación insuficiente	5	4	<b>38</b>	18	<b>36</b>	3.78	<b>4</b>	1.13
Falta de herramientas de soporte para el mantenimiento	14	9	24	27	26	3.41	<b>4</b>	<b>1.35</b>

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Desafíos en las prácticas de la ingeniería del software.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación con las actividades de mantenimiento.

Los desafíos generales que indican los participantes son: cuáles son las mejores prácticas de gestión de mantenimiento del software, cómo realizar los procesos de *clean code* y *refactoring*, cuáles son las mejores prácticas para la corrección del código, cómo mejorar el acoplamiento y la cohesión de código de manera segura, cómo lidiar con la complejidad del código, cómo identificar el código obsoleto y qué hacer con él, cómo mejorar la productividad en las labores de mantenimiento, cómo realizar la trazabilidad entre pruebas, código y requerimientos, cómo mejorar la calidad de la documentación interna, la documentación del sistema y cuáles son los estándares recomendados, cómo tomar la decisión sobre a cual código se le debe realizar el mantenimiento preventivo o adaptativo y cómo pasar de mantenimiento reactivo a preventivo. Cómo minimizar el impacto de los diferentes estilos de programación, cuáles son las mejores herramientas para el soporte y para el mantenimiento y cómo integrar los equipos de calidad en las labores de mantenimiento.

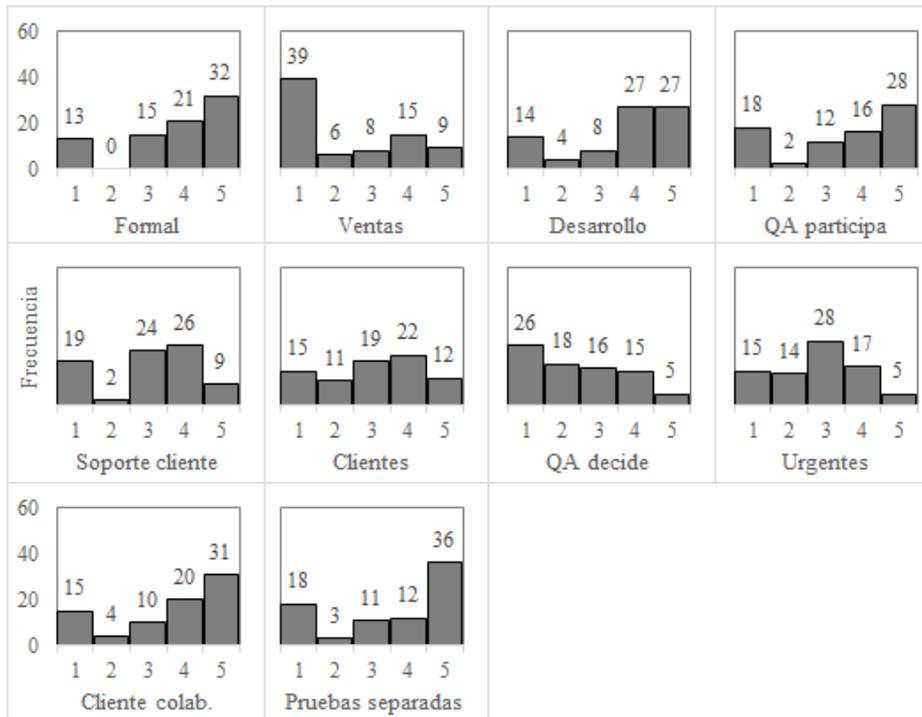
### 5.3.6 Administración de la configuración, liberaciones y soporte

**Prácticas durante la liberación del producto.** En esta pregunta se investigan las prácticas relacionadas con la liberación del producto. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si tienen un único punto formal para realizar las liberaciones de *software*, los representantes que participan en las decisiones sobre la liberación del producto (marketing, desarrollo, pruebas (QA), soporte al cliente, los clientes). Se consulta si solo QA toma las decisiones de cuando liberar, si liberan versiones del *software* de manera urgente (parches) sin realizar pruebas exhaustivas, si existe una colaboración estrecha con el cliente cuando se realizan las liberaciones y/o si se tiene un entorno de pruebas y de liberación separados.

Tal como se muestra en la Figura 35 y el Cuadro 41, las prácticas más frecuentes son: tener un único punto formal para realizar las liberaciones de software, representantes de desarrollo participan en las decisiones de cuando liberar y representantes de pruebas (QA) participan en las decisiones de cuando liberar, respectivamente. Contrario

a lo reportado por [GAR15], en nuestro estudio no es común que solo QA tome las decisiones de cuando liberar. Además, en nuestro estudio se reporta la colaboración estrecha con el cliente cuando se realizan liberaciones y el contar con un entorno de pruebas y de liberación separados.



**Figura 35 Frecuencia de respuestas para en administración de liberaciones**

**Cuadro 41 Administración de liberaciones**

Administración de liberaciones	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Tienen un único punto formal para realizar las liberaciones	16	0	19	26	<b>40</b>	<b>3.73</b>	<b>4</b>	1.41
Representantes de marketing participan en las decisiones de cuando liberar	<b>51</b>	8	10	19	12	2.34	1	1.54
Representantes de desarrollo participan en las decisiones de cuando liberar	18	5	10	<b>34</b>	<b>34</b>	<b>3.61</b>	<b>4</b>	1.45
Representantes de pruebas (QA) participan en las decisiones de cuando liberar	24	3	16	21	<b>37</b>	3.45	<b>4</b>	<b>1.58</b>
Representantes de soporte al cliente participan en las decisiones de cuando liberar	24	3	<b>30</b>	<b>33</b>	11	3.05	3	1.33
Los clientes participan en el proceso de liberación	19	14	24	<b>28</b>	15	3.06	3	1.34
Solo QA toma las decisiones de cuando liberar	<b>33</b>	23	20	19	6	2.44	2	1.29
Libran software urgente sin pruebas exhaustivas	19	18	<b>35</b>	22	6	2.78	3	1.17
Colaboran estrechamente con el cliente cuando se realizan liberaciones	19	5	13	25	<b>39</b>	3.60	<b>4</b>	1.51
Se tiene un entorno de pruebas y de liberación separados	23	4	14	15	<b>45</b>	3.56	<b>4</b>	<b>1.61</b>

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Prácticas durante el soporte del producto.** En esta pregunta se investigan las prácticas relacionadas con las actividades de soporte. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si tienen un equipo dedicado a soporte, si proveen soporte en diferentes niveles ("*tiered support*"), si los equipos de soporte, desarrolladores y QA trabajan en estrecha colaboración, si utilizan alguna herramienta para el seguimiento de los reportes del cliente ("*tickets*"), si la herramienta de "*issues*" es diferente a la de seguimiento de defectos, si el equipo de soporte recibe entrenamiento formal del producto, si se asigna a cada cliente importante una persona dedicada al soporte y/o si los desarrolladores trabajan con los clientes en el lugar ("*on-site*").

Los resultados se muestran en la Figura 36 y el Cuadro 42. Las prácticas más frecuentes son: tener un equipo dedicado a soporte y usar una herramienta para el seguimiento de los reportes del cliente ("*tickets*"). La práctica menos frecuente es la asignación a cada cliente importante de una persona dedicada a soporte. Los resultados son similares a los reportados por [GAR15].

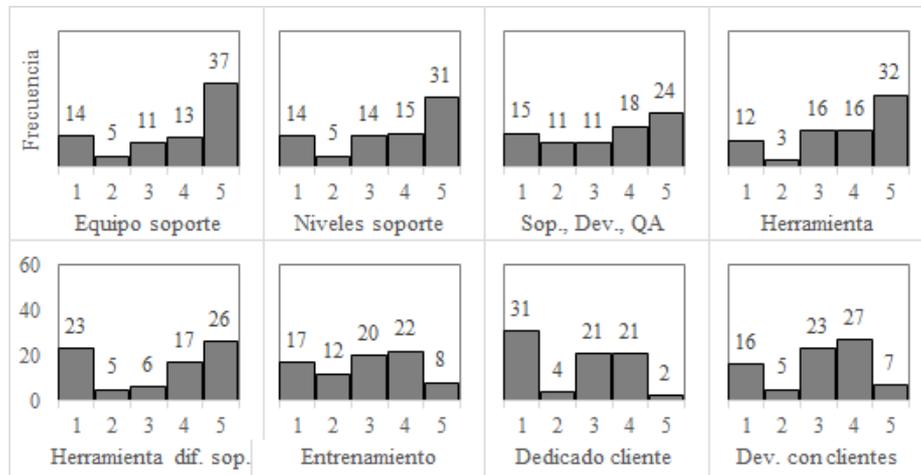


Figura 36 Frecuencia de respuestas para actividades de soporte

Cuadro 42 Actividades de soporte

Actividades de soporte	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Tienen un equipo dedicado a soporte	18	6	14	16	46	3.68	4	1.53
Proveen soporte en diferentes niveles ("tiered support")	18	6	18	19	39	3.56	4	1.50
El equipo de soporte, desarrolladores y QA trabajan en estrecha colaboración	19	14	14	23	30	3.32	4	1.51
Usamos una herramienta para el seguimiento de los reportes del cliente ( <i>tickets</i> )	15	4	20	20	41	3.67	4	1.43
La herramienta de <i>issues</i> es diferente a la de seguimiento de defectos	30	6	8	22	34	3.23	4	1.68
El equipo de soporte recibe entrenamiento formal del producto	22	15	25	28	10	2.90	3	1.31
Se asigna a cada cliente importante una persona dedicada a soporte	39	5	27	27	3	2.48	3	1.32
Los desarrolladores trabajan con los clientes en el lugar (" <i>on-site</i> ")	21	6	29	35	9	3.05	3	1.27

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Desafíos en las prácticas de la ingeniería del software.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación con las actividades de administración de la configuración.

Los desafíos generales que indican los participantes son: cuáles son las mejores prácticas en la gestión de configuración y sus diferentes ambientes de desarrollo, las mejores prácticas en la gestión de versiones para la agilización de despliegues, mejores prácticas en los procesos de entrega continua y cómo gestionar las dependencias entre componentes y lograr la automatización en las liberaciones y la generación de los *builds*, cuáles son las mejores prácticas para la toma de la decisión de cuándo liberar.

### 5.3.7 Administración de proyectos de *software*

**Medición y estimación en la administración de proyectos.** En esta pregunta se consulta sobre el uso de mediciones y estimaciones en los proyectos. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si las decisiones tomadas por los administradores de proyectos son basadas en métricas y mediciones sistemáticas, si la estimación de esfuerzo y costos se realiza de forma sistemática antes de cada proyecto, si utilizan alguno de los métodos formales de estimación de esfuerzo y costos (por ejemplo COCOMO) y/o si utilizan el juicio experto para la estimación de esfuerzo y costos.

De acuerdo a los resultados mostrados en la Figura 37 y el Cuadro 43, el uso del juicio experto para la estimación de esfuerzo y costos es una práctica común y la utilización de métodos formales de estimación es limitada. Al igual que lo reportado en [GAR15], aunque existe algún nivel de adopción en la práctica de realizar la estimación de esfuerzo y costos de forma sistemática antes de cada proyecto, muy pocos lo hacen mediante los métodos formales tradicionales. Solo un participante de este estudio reporta el uso de COCOMO. Las metodologías reportadas para estimación son el uso de historias de usuario y los puntos de casos de uso aplicado a historias de usuario.

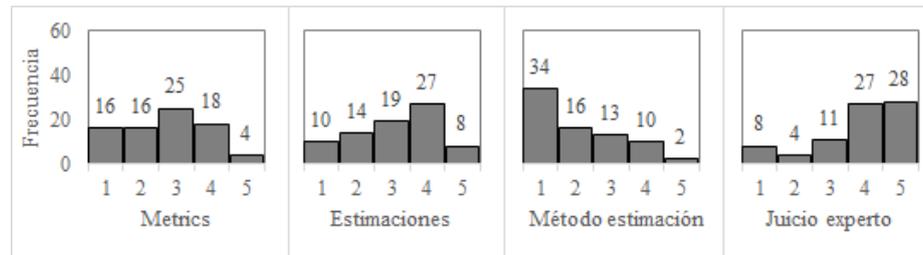


Figura 37 Frecuencia de respuestas para las prácticas de medición y estimación

**Cuadro 43 Prácticas de medición y estimación**

Prácticas en proyectos	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Las decisiones tomadas por los administradores de proyectos son basadas en métricas y mediciones sistemáticas	20	20	<b>32</b>	23	5	2.72	3	1.18
La estimación de esfuerzo y costos se realiza de forma sistemática antes de cada proyecto	13	18	24	<b>35</b>	10	3.12	3	1.21
Utilizan métodos formales de estimación de esfuerzo y costos (por ejemplo COCOMO)	<b>45</b>	21	17	13	3	2.07	2	1.19
Utilizan juicio experto para la estimación de esfuerzo y costos	10	5	14	<b>35</b>	<b>36</b>	<b>3.81</b>	<b>4</b>	<b>1.27</b>

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Planificación y seguimiento de proyectos.** En esta pregunta se consulta sobre las prácticas de planificación y seguimiento de proyectos. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre si en los proyectos se crean los planes operativos semanalmente, si se crean planes para todo el proyecto, pero no se ajustan a estándares definidos, si se crean planes para todo el proyecto basado en un proceso estándar definido, si en los proyectos el equipo de gestión crea informes de progreso periódicos (semanal, mensual) y/o si realizan reuniones de avance semanal o mensual.

De acuerdo a los resultados mostrados en la Figura 38 y el Cuadro 44, al igual que lo reportado en [GAR15], la práctica más comúnmente utilizada es la de realizar reuniones periódicas de avance (semanal, mensual). Además, se reporte que en los proyectos, el equipo de gestión crea informes de progreso periódicos.

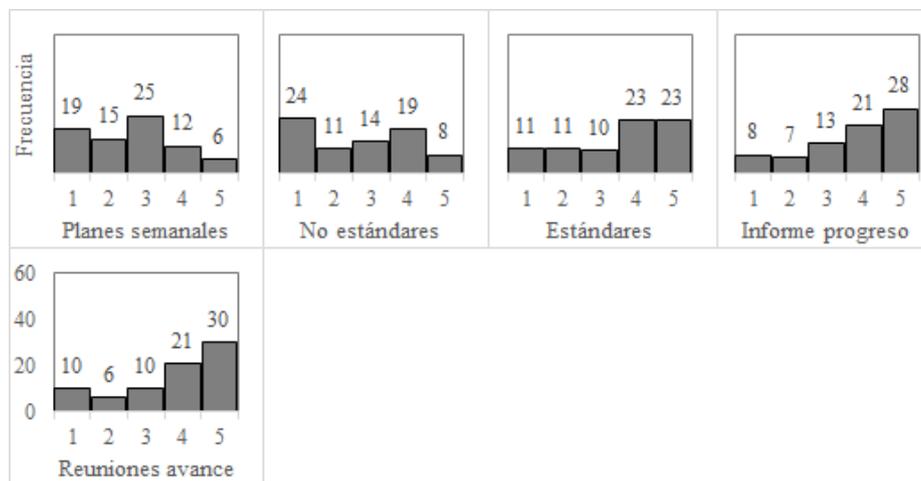


Figura 38 Frecuencia de respuestas en la planificación y seguimiento

Cuadro 44 Planificación y seguimiento

Actividades de planificación	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Creamos planes operativos semanales	25	19	32	16	8	2.62	3	1.24
Creamos planes para todo el proyecto, no nos ajustamos a estándares definidos	32	14	18	25	11	2.68	3	1.42
Creamos planes para todo el proyecto basado en un proceso estándar definido	14	14	13	29	29	3.46	4	1.41
En los proyectos, el equipo de gestión crea informes de progreso semanal, mensual	10	9	17	27	36	3.70	4	1.33
Realizamos reuniones de avance semanal, mensual	13	8	13	27	39	3.71	4	1.39

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Desafíos en las prácticas de la ingeniería del *software*.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación con las actividades de administración de la proyectos.

Los desafíos generales que indican los participantes son: cuáles son las mejores prácticas esenciales para la administración de proyectos en diferentes metodologías de desarrollo, cómo empatar un manejo tradicional de proyectos (dado por la oficina de proyectos) con las prácticas ágiles de los equipos, cuáles son las mejores prácticas para la gestión de proyectos cuando la organización jerárquica de la institución no está por proyectos, cuáles son las mejores prácticas para garantizar el apoyo de la alta gerencia a los proyectos y cómo maximizar el aprovechamiento de la disponibilidad del usuario

final, cuáles son las prácticas para realizar la gestión de cambios fuera del plan original, para la gestión de riesgos y la administración de recursos, cómo mejorar el trabajo en equipo. Además, cómo realizar de manera efectiva el proceso de estimación de tiempo y costo de desarrollo, cómo establecer métricas orientadas a la producción (rendimiento) en el desarrollo de software.

### 5.3.8 Aseguramiento de la calidad del *software*

**Prácticas de aseguramiento de calidad.** En esta pregunta se consulta sobre las prácticas SQA. La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Los participantes son consultados sobre el seguimiento de especificaciones de calidad de un estándar de la industria (ISO / IEC 9126, DO-178B, IEEE 730-1998, otro), la definición clara del objetivo del proceso de verificación y validación (la densidad de defectos de nuestros proyectos), si se llevan a cabo revisiones por pares sistemáticas y auditorías en los proyectos y/o si se tienen enfoques bien definidos para la gestión de defectos (severidad, tipos de defectos, entre otros).

En nuestro estudio, los participantes reportan tener enfoques bien definidos para la gestión de defectos (severidad, tipos de defectos, entre otros), pero no es común la adopción de especificaciones de calidad de un estándar, tal como se muestra en la Figura 39 y el Cuadro 45. En nuestro estudio, las prácticas relacionadas son SQA tienen una adopción menor que a las reportadas en [GAR15].

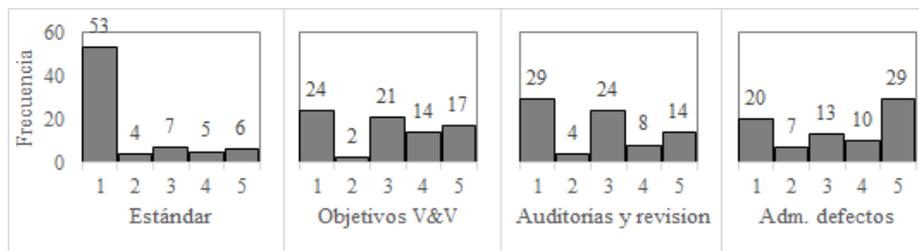


Figura 39 Frecuencia de respuestas para las prácticas de aseguramiento de la calidad

**Cuadro 45 Prácticas de aseguramiento de la calidad**

Aseguramiento de la calidad	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Siguen las especificaciones de calidad de un estándar de la industria (ISO / IEC 9126, DO-178B, IEEE 730-1998, otro)	71	5	9	7	8	1.76	1	1.32
Tienen bien definido el objetivo del proceso de verificación y validación (la densidad de defectos de nuestros proyectos)	31	3	27	18	22	2.97	3	1.53
Se llevan a cabo revisiones por pares sistemáticas y auditorias en los proyectos	37	5	30	10	18	2.67	3	1.50
Tienen enfoques bien definidos para la gestión de defectos (severidad, tipos de defectos, entre otros)	25	9	16	13	37	3.27	3	1.63

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

#### 5.4 Herramientas de la ingeniería del *software* (SE)

La RQ4 investiga las herramientas que soportan las actividades de las prácticas de SE. Es interesante conocer el nivel de adopción de estas herramientas en la industria.

Los participantes son consultados sobre el uso de herramientas para requerimientos (ej. documentación de requerimientos), diseño (ej. modelaje UML), pruebas (ej. pruebas automáticas GUI), mantenimiento del *software* (ej. comprensión o la reingeniería de los artefactos de diseño), para el modelado de procesos y gestión de procesos y calidad de *software* (ej. análisis de código estático). Además, si utilizan herramientas para el versionamiento, liberaciones y “*builts*”, y finalmente, herramientas para la administración (ej. planificación, seguimiento, gestión y riesgos). La pregunta se plantea en una escala de Likert con las opciones: 1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre.

Al igual que lo reportado en [GAR15], las herramientas para la gestión de la configuración (versionamiento, liberaciones y “*builts*”) y las herramientas para la administración (planificación, seguimiento, gestión y riesgos) son las más comúnmente utilizadas. Las herramientas menos utilizadas son las de mantenimiento del *software* (comprensión o la reingeniería de los artefactos de diseño) y para la calidad de *software* (análisis de código estático), tal como se muestra en la Figura 40 y el Cuadro 46.

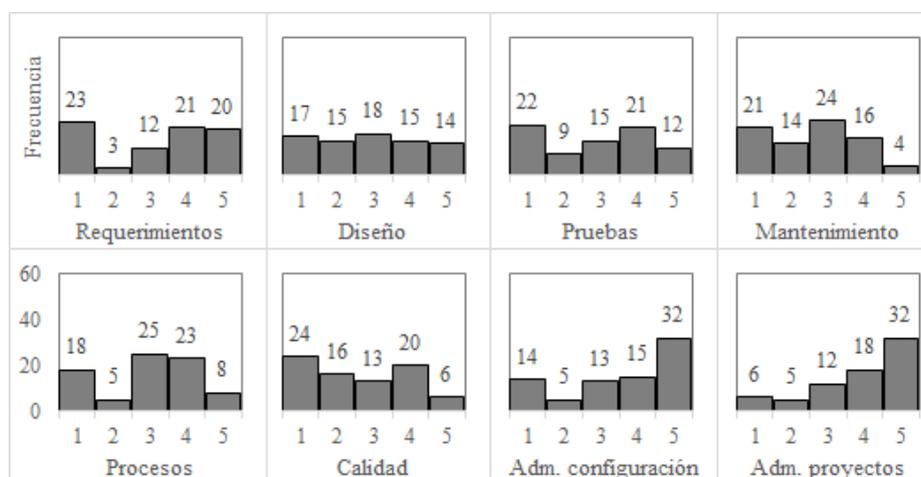


Figura 40 Frecuencia de respuestas para las herramientas

Cuadro 46 Herramientas

Herramientas utilizadas	Porcentajes (%)					Pm	Md	SD
	1	2	3	4	5			
Herramientas para requerimientos (documentación de requerimientos)	29	4	15	27	25	3.15	4	1.58
Herramientas para diseño (modelado UML)	22	19	23	19	18	2.92	3	1.40
Herramientas para pruebas (pruebas automáticas GUI)	28	11	19	27	15	2.90	3	1.46
Herramientas para mantenimiento del software (comprensión o la reingeniería de los artefactos de diseño)	27	18	30	20	5	2.59	3	1.22
Herramientas para el proceso de ingeniería (modelado de procesos y gestión de procesos)	23	6	32	29	10	2.97	3	1.30
Herramientas para la calidad de software (análisis de código estático)	30	20	16	25	8	2.59	2	1.35
Utilizamos las herramientas de Software (versionamiento, <i>releases</i> y <i>builds</i> )	18	6	16	19	41	3.58	4	1.51
Herramientas para la administración (planificación, seguimiento, gestión y riesgos)	8	7	16	25	44	3.89	4	1.28

1=Nunca 2=Casi nunca 3=A veces 4=Casi siempre 5=Siempre

**Herramientas utilizadas.** Se consulta a los profesionales cuáles son las mejores herramientas de producción de software en la prácticas de administración, desarrollo, pruebas, medición, entre otras utilizadas en los procesos de desarrollo de *software*.

Las herramientas mencionadas por los profesionales son: *Microsoft Visual Studio .NET*, *Microsoft Test Manager*, *Microsoft Team Foundation Server*, *Microsoft Project*, *Sharepoint*, *Software AG's Natural Application Development Platform*, herramientas de código abierto para diagramas UML, *CruiseControl.net*, *Java Eclipse*, *Kanban*, *Jira Service Desk*, *Jenkins*, *Gerrit Code Review*, *Git*, *IBM Rational ClearQuest*, entre otras. Algunas de las herramientas de son desarrolladas a lo interno de las organizaciones y utilizadas para la automatización de la integración continua, automatización de la gestión de dependencias entre componentes, liberaciones automáticas y seguimiento de versiones.

**Desafíos en las prácticas de la ingeniería del *software*.** Se consulta a los participantes sobre cuáles son los desafíos o retos a los que se enfrentan en su práctica profesional en relación a los métodos y herramientas.

Los desafíos generales que indican los participantes son: cuáles son las mejores prácticas para facilitar los procesos de adopción de nuevas tecnologías y reducir la resistencia al cambio, cuál es la mejor manera de realizar el proceso de migración a nuevas herramientas, cómo se debe realizar el proceso de adopción de nuevas metodologías y tecnologías en equipos consolidados, cómo hacer la transición desde las metodologías tradicionales hacia metodologías ágiles y cuáles son los mejores artefactos esenciales que aporten valor en el ciclo de vida de desarrollo, cómo realizar el proceso de migración a tecnologías y herramientas *open source*, como configurar la integración entre diferentes herramientas y cómo adoptar el uso de las herramientas en la nube. Cuáles son las mejores herramientas para la especificación de requerimientos y diseño que automatizan parte del proceso de desarrollo y las mejores herramientas integradas para el proceso de desarrollo de software y de análisis de código. Finalmente, cuales son las mejores prácticas para realizar un proceso de migración continua de aplicaciones hacia las nuevas tecnologías.

## 5.5 Investigación y colaboración con la academia

La RQ5 busca determinar las actividades de investigación realizadas por las organizaciones, el nivel de interacción con investigadores académicos y literatura científica. Finalmente, intenta identificar algunos de los principales desafíos asociados a las prácticas de la SE con los que se enfrentan las organizaciones.

En nuestro estudio, solo cinco de 136 organizaciones indican que cuentan con una unidad de investigación (4%). Los resultados distan de los reportados por [GAR15]. Estos resultados indican la necesidad de que las organizaciones de desarrollo puedan destinar recursos a la investigación sobre cómo mejorar sus prácticas en SE.

**Interacción con investigadores académicos y literatura científica.** Tal como se observa en la Figura 41 y el Cuadro 47, los participantes reportan poca interacción con investigadores académicos. En cuanto a la lectura de artículos científicos la tendencia

mejora un poco con respecto a la pregunta anterior. Los resultados son similares a las reportadas en [GAR15]. Es necesario incentivar las colaboraciones industria y academia aprovechando los incentivos que el gobierno pueda otorgar para estos efectos.

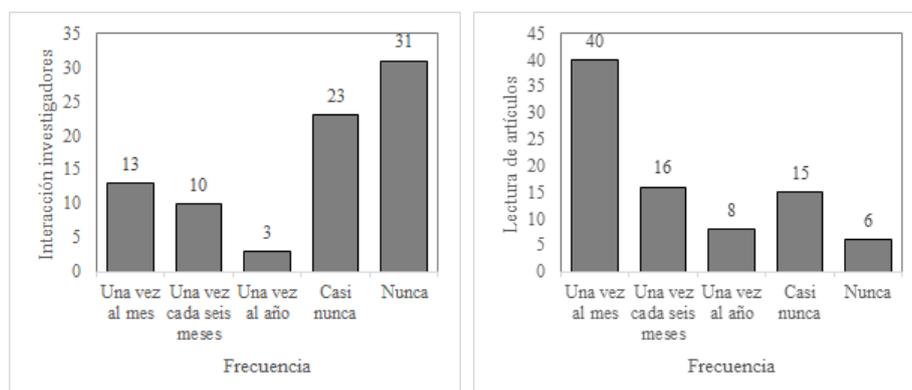


Figura 41 Frecuencia de respuestas para la interacción con la academia

Cuadro 47 Interacción con la academia

Investigadores	n	%	Artículos	n	%
Una vez al mes	13	16%	Una vez al mes	40	47%
Una vez cada seis meses	10	13%	Una vez cada seis meses	16	19%
Una vez al año	3	4%	Una vez al año	8	9%
Casi nunca	23	29%	Casi nunca	15	18%
Nunca	31	39%	Nunca	6	7%
<b>Total</b>	<b>80</b>	<b>100%</b>	<b>Total</b>	<b>85</b>	<b>100%</b>

Al consultar a los participantes sobre cual creen que es la mejor forma para que las universidades impulsen espacios de diálogo entre académicos y profesionales donde compartir conocimiento científico, indican que se deben realizar reuniones, charlas, conversatorios, mesas redondas, grupos focales, talleres y demostraciones, publicaciones y boletines técnicos, congresos y conferencias, foros virtuales, tutoriales en video, visitas a las organizaciones, programas de intercambio de conocimiento y proyectos de colaboración, intercambio de estudiantes con las organizaciones y pasantías para los profesionales. En el caso del sector público consideran que la participación de las universidades en proyectos de desarrollo de soluciones integrales es clave. Los participantes opinan que se deben mejorar los canales de comunicación para anunciar los eventos y que la academia debe ser más abierta, con visitas a las organizaciones, realizando procesos de investigación aplicada que generen valor agregado.

Al consultar a los participantes sobre si consideran que la literatura científica proporciona un valor agregado para sus actividades profesionales, indican que si se percibe el valor agregado en su práctica profesional ya que proporcionan información confiable

y actualizada. Además, reconocen que la literatura juega un papel importante en los procesos de mejora continua.

Por otro lado, los profesionales explican que algunas veces los artículos no se apegan a la realidad o son difíciles de poner en práctica. Muchas veces la fuente de información utilizada por los profesionales son los foros y wikis reconocidos en la comunidad tecnológica y que permiten de manera más directa encontrar el "how to" para problemas específicos. Finalmente, indican que aunque la literatura científica ofrece prácticas novedosas estas son difíciles de aplicar al entorno profesional dado que las organizaciones son cautelosas ante los cambios. Ellos indican que si las investigaciones demuestran una tendencia a resultados exitosos en relación a la adopción de algunas prácticas, esto sería muy valioso para los profesionales, ya que se requiere garantizar que las organizaciones puedan lograr los resultados exitosos reportados en las investigaciones. Finalmente, indican que muchas veces no es fácil el acceso a la literatura científica por las restricciones de derechos de propiedad intelectual de las bases de datos.

**Desafíos en la ingeniería del software.** Se consulta a los participantes sobre cuáles son los desafíos que desearían que la academia investigue o en la cuáles se requirieran colaboraciones industria academia.

El detalle de los retos en las prácticas por cada una de las fases y actividades del ciclo de vida de desarrollo se detalla en cada sección del reporte. El resumen de las respuestas proporcionadas por los participantes son las siguientes:

- Cómo garantizar la calidad del código fuente.
- Cuáles son las métricas que generan más valor en el ciclo de vida de desarrollo y como pueden ser automatizadas.
- Cómo estimar el tiempo y costo de manera eficiente.
- Cuáles son los artefactos esenciales que generan más valor en el ciclo de vida de desarrollo.
- Cuáles son las mejores prácticas para la capacitación de personal en nuevas metodologías y herramientas.
- Cómo desarrollar las habilidades requeridas en diseño, programación y pruebas en los nuevos profesionales.

Los retos planteados por los profesionales en su práctica profesional pueden ser utilizados como una hoja de ruta para futuras investigaciones o colaboraciones industria academia. De acuerdo a los resultados, aunque algunos de los retos planteados ya han sido discutidos en la literatura científica, existen brechas entre la literatura científica y su aplicación en la práctica profesional que deben ser reducidas.

## 6 Discusión

En esta sección se presentan el resumen de los resultados de la encuesta y se discuten las lecciones aprendidas. Finalmente, se detallan las principales limitaciones del estudio.

## 6.1 Resumen de resultados

Nuestro estudio analiza las respuestas de 278 profesionales de la ingeniería del *software* trabajando en organizaciones de desarrollo costarricenses. La encuesta recoge las opiniones de profesionales de diferentes perfiles relacionados con el desarrollo de *software*, distintos rangos de edad, años de experiencia profesional, grado académico y distintas universidades. Además, los participantes pertenecen a distintos tipos de organización, de distintos tamaños y que desarrollan proyectos para diferentes sectores. La combinación de los datos recolectados permite contar con un panorama general sobre la adopción de las prácticas de la ingeniería de software en nuestro país.

La encuesta revela información interesante sobre las diferentes tendencias en la industria de *software* en Costa Rica. Además, de los principales desafíos a los que se enfrenta la comunidad de la ingeniería de *software* en el país, que incluye la academia y la industria privada y pública.

El resumen de las principales características del perfil de los profesionales se lista a continuación:

- La mayoría de los profesionales en desempeñan labores como desarrolladores de software y programadores e ingenieros de software. Además de perfiles relacionados con la práctica ingenieril tales como requerimientos y diseño.
- La mayoría de los profesionales en puestos docentes trabajan en la industria (80%) y en todos los casos forman profesionales en la SE.
- El 39% de los participantes reportan desempeñar solo una posición en sus organizaciones. Sin embargo 84 participantes (30%) reportan desempeñarse en 4 o más posiciones.
- La mayoría de los profesionales se encuentran trabajando en organizaciones nacionales grandes (501+ empleados) y que su giro de negocio no es el desarrollo de software.
- El promedio de experiencia reportado es de 12.9 años y la media de 12 años. El 47% de los participantes tienen una experiencia menor o igual a 10 años. Alrededor del 23% de los participantes tiene 5 o menos años de experiencia, el 24% entre 5 y 10 años de experiencia y el 53% más de 10 años. El 76% de los participantes son masculinos y el 21% femeninas.
- El 40% de los participantes cuentan con una maestría, el 22% con una licenciatura y el 26% con el bachillerato. Además, el 49% de los participantes se encuentra estudiando un grado superior o una actualización técnica.
- La formación universitaria es ingeniería en computación (33%), sistemas de información (22%), ciencias de la computación (18%) e ingeniería del *software* (14%).
- La mayoría de los entrevistados estudió en las universidades públicas costarricenses (66%).
- La mayoría de los participantes trabajan en San José (69%), Alajuela (10%), Heredia (10%) y Cartago (6%).
- El sector objetivo de los productos de software son el gobierno y sector público (30%), la banca y finanzas (16%) y comercio electrónico en general (11%).
- Los participantes perteneces a organizaciones del sector gobierno (24%), la industria de desarrollo de software a la medida (14%) y banca y finanzas (11%).

- La mayoría de las organizaciones de los participantes son de capital nacional (67%), el 12% de las organizaciones son de capital extranjero y el 10% de ambos.
- El 64% de los participantes pertenecen a organizaciones grandes (501+ empleados) y los tipos de clientes para los que trabajan son internos (27%), externos (13%) o ambos (42%).
- Los proyectos son en su mayoría de software a la medida a partir de las necesidades particulares de los clientes (36%) y desarrollados "In-house" (34%).
- El 45% de los participantes realizan proyectos de desarrollo, validación, verificación y mantenimiento de *software*, el 34% llevan a cabo proyectos de desarrollo y mantenimiento y el 2% indica que realizan solo proyectos de validación y verificación.

El resumen de los principales hallazgos sobre las prácticas de la ingeniería del software se resume a continuación:

- Los participantes reportan participación activa en las fases de requerimientos, diseño, desarrollo, pruebas y mantenimiento. Además reportan su participación en el proceso de administración de proyectos.
- Las fases con menor nivel de participación son las de administración de la configuración, definición, evaluación y medición de los procesos de ingeniería del *software*, aseguramiento de la calidad y documentación técnica.
- Los participantes desarrolladores reportan un ligero predominio de las fases de desarrollo y mantenimiento de *software*.
- Los participantes reportan sus principales desafíos en las actividades de aseguramiento de calidad y comunicación con la alta administración y con los usuarios finales. La fase más desafiante es la de pruebas.
- Las metodologías de desarrollo más utilizadas son las ágiles (46.7%), principalmente *Scrum* con un 24.5%, el 19.4% indican seguir las prácticas ágiles sin adoptar un método específico. Las metodologías tradicionales como cascada (19.9%), prototipos (19.0%), incremental (10.2%) y espiral (3.7%) aún se mantienen vigentes.
- El 85 % de los profesionales usan o una metodología ágil o la metodología de cascada pero no ambas, esto indica que si una organización utiliza metodologías ágiles no utiliza cascada y viceversa.
- Existe una baja adopción de modelos de calidad. Las organizaciones que adoptaron algún modelo utilizan el ISO 9000 ("*Quality management systems*") (41%), el CMMI ("*Capability Maturity Model Integration*") (36%) y el otros como COBIT, ITIL, ISO 27001, ISO 42010 e ISO 19510 (23%).
- Aunque la mayoría de las organizaciones no reportan la adopción formal de modelos o estándares, la tendencia indica la aplicación de tareas basadas en un estándar de procesos de un manual de calidad, se realizan tareas definidas por el equipo de administración de proyectos y el monitoreo y evaluación sistemática del proceso de desarrollo.
- Las principales prácticas en requerimientos son la documentación y revisión formal de todas las características (*features*) del producto, la validación de los requerimientos mediante inspecciones, revisión de pares y *walk throughs*.

- Las notaciones más utilizadas para requerimientos son casos de uso y escenarios (27%), de "*User stories* y *story boards*" (22.8%), el uso de diagramas de actividad UML (18.6%) y el uso lenguaje natural (15.8). Solo el 0.9% indica que no se documentan requerimientos.
- Las principales prácticas en diseño son la asignación de personal dedicado a la arquitectura de software y/o el diseño de alto nivel y el equipo de trabajo realiza revisiones informales del diseño.
- Las principales tareas realizadas son el diseño de clases, componentes e interfaz de usuario, el diseño por medio de prototipos, y el diseño de los servicios, la arquitectura y el producto.
- Los atributos de calidad del diseño como la seguridad, usabilidad, extensibilidad, tolerancia a fallos, mantenibilidad, modularidad, confiabilidad, reusabilidad y robustez son considerados muy importantes en el diseño.
- Las prácticas más comunes en desarrollo son que los desarrolladores son responsables por el diseño de alto nivel y la implementación, el diseño y la codificación son realizados en conjunto, y el proceso de los "*builts*" está documentado y se le da seguimiento.
- El 46% de los participantes reportan el uso de más de un lenguaje de programación. Los lenguajes más utilizados son los de la familia .Net (C#, VB, ...) con un 48%, seguido de Java con un 26%. C/C++ tienen un porcentaje del 9%, PL/SQL Oracle un 7.9% y PHP con un 5.5%.
- Las prácticas más comunes en pruebas son que un gerente, experto en el dominio del negocio (cliente o apoyo) ayuda a probar el producto, los desarrolladores prueban el producto antes de cada liberación y el uso de las pruebas de unidad ("*unit tests for code*"), 25% de los participantes indican que siempre se tiene un equipo separado para realizar las pruebas y el 29% que siempre las nuevas características ("*features*") son probadas independientemente por un equipo de pruebas.
- La mayoría de los participantes realizan la fase de pruebas, de manera tradicional, una vez que el *software* está totalmente desarrollado ("*Test last Development*").
- Los tipos de pruebas más populares son las pruebas de aceptación de usuario, pruebas funcionales, pruebas de sistema y pruebas de integración.
- Para el diseño de pruebas, las técnicas basadas en modelos son las más utilizadas (22%), análisis de valores límite (14.4%), caja blanca (14.4%), clases de equivalencia (9.8%) y caja negra (5.3%). La no utilización de una técnica específica es alta (34.1%).
- El uso de pruebas manuales prevalece sobre las automatizadas. El 33% de los participantes dicen utilizar pruebas automatizadas siempre o casi siempre.
- La utilización de métricas de cobertura de código y las métricas de calidad es limitada en la industria. Las métricas de calidad más frecuentes reportadas son los casos de pruebas de aceptación aprobados y los defectos detectados por un periodo de tiempo.
- La tendencia principal de la relación entre *testers* y desarrolladores es de 1:5+ (24%). Gran cantidad de participantes indican que las organizaciones no hacen una distinción entre *testers* y desarrolladores o no realizan ninguna medición al respecto.

- El criterio más utilizado para la finalización de pruebas es el de ejecutar todos los casos de prueba hasta no encontrar más defectos. La adopción de métricas de cobertura se reporta como de uso limitado en nuestra industria.
- El tipo de mantenimiento correctivo es la actividad más común y el mantenimiento preventivo la menos utilizada. Los participantes reportan que las actividades de mantenimiento preventivo y adaptativo son los más difíciles de realizar. Los diferentes estilos de programación, la documentación insuficiente, y la falta de herramientas de soporte para el mantenimiento son las principales dificultades.
- Las prácticas durante la liberación del producto son el tener un único punto formal para realizar las liberaciones de software, donde representantes de desarrollo y pruebas (QA) participan en las decisiones de cuando liberar. Además se reporta la colaboración estrecha con el cliente cuando se realizan liberaciones.
- Las prácticas más frecuentes en soporte son el tener un equipo dedicado a soporte y usar una herramienta para el seguimiento de los reportes del cliente (“*tickets*”).
- En cuanto a estimación es una práctica común el uso del juicio experto. La utilización de métodos formales de estimación es limitada.
- Para la planificación y seguimiento de proyectos las prácticas más comunes son las reuniones periódicas de avance (semanal, mensual) y la creación de informes de progreso periódicos por parte del equipo de gestión.
- Las principales prácticas de SQA reportadas son que se tienen enfoques bien definidos para la gestión de defectos (severidad, tipos de defectos, entre otros), pero no es común la adopción de especificaciones de calidad de un estándar. En nuestro estudio, las prácticas relacionadas son SQA reportan una adopción limitada.
- Las herramientas más comúnmente utilizadas son las de la gestión de la configuración (versionamiento, liberaciones y “*builts*”) y las herramientas para la administración (planificación, seguimiento, gestión y riesgos). Las menos utilizadas son las de mantenimiento del software (comprensión o la reingeniería de los artefactos de diseño) y para la calidad de *software* (análisis de código estático).
- Los participantes reportan poca interacción con investigadores académicos, sin embargo, reportan la lectura de artículos científicos.

Finalmente, la mayoría de los retos a los que enfrentan los profesionales en la práctica profesional han sido discutidos en la literatura científica pero aún existe una brecha entre la literatura científica y su aplicación en la práctica profesional. Los retos planteados por los profesionales deben ser los motivantes para impulsar investigaciones y colaboraciones entre la industria y la academia y deben ser atacados en distintos niveles. Las guías de las mejores prácticas en el área como lo es el SWEBOK [BOU14], pueden ser la base para que los académicos y profesionales puedan hablar un lenguaje común en que se puedan implementar iniciativas para mejorar las prácticas adoptadas por la industria. La academia debe ser más efectiva en los procesos de transferencia de conocimientos para que puedan ser aplicados por los profesionales, y de igual manera, es responsabilidad de los profesionales mantenerse actualizados con la literatura clave en el área y así adoptar las mejores prácticas conocidas.

## 6.2 Lecciones aprendidas

La experiencia al ejecutar la encuesta confirma las recomendaciones realizadas en estudios previos sobre este tipo de estudios [PUN03] [KAN13] [GAR15], entre otros.

Al igual que en estudios previos, se realizó una categorización de las preguntas de acuerdo al SWEBOK (2004) y revisado contra [BOU14], posteriormente se validó el instrumento con profesionales para asegurar que la terminología era la común en la industria. El instrumento se aplica en el lenguaje español y los investigadores fueron los responsables de realizar las traducciones respectivas.

Todos los participantes fueron invitados a participar de manera anónima y voluntaria por correos de invitación enviados por medio de distintas instituciones relacionadas con la industria de *software* en Costa Rica. Dados los distintos perfiles, se les indicó a los participantes que respondieran las secciones que consideran de acuerdo a sus áreas de experticia.

El medio de contacto más efectivo fueron las Universidades Públicas, seguido por las Instituciones del Sector Público. Los participantes que fueron contactados por medio de las universidades tienen una tasa de finalización de respuestas del 40% y los de las instituciones públicas del 61%. Los desarrolladores e ingenieros de requerimientos tienen una tasa de finalización de entre 40-43%, los analistas o ingenieros de software, arquitectos y diseñadores del 33-34%, los analistas de negocio y líderes de proyecto del 30%, los administradores de proyectos del 25% y los gerentes de alto nivel (CEO, CIO, CFO) e ingenieros de calidad del 12-13%. Los profesionales que tienen alguna relación con labores docentes tienen una tasa de finalización de respuestas del 38% y los relacionados con labores de investigación del 10%.

La participación en la encuesta fue anónima y los resultados publicados son mostrados de manera agregada, por lo que se espera que en futuros cuestionarios los participantes se sientan motivados a responder la totalidad del cuestionario al observar que los datos reportados no afectan la privacidad de sus organizaciones.

En nuestro entendimiento, este es el primer estudio de este tipo que se realiza en el país. Esperamos que los datos recopilados generen valor agregado a las organizaciones y motive su participación futura.

Similar a experiencias reportadas en estudios anteriores, algunos participantes dejaron preguntas sin responder y otros abandonaron el cuestionario sin terminarlo. Todas las respuestas parciales fueron consideradas para el análisis de resultados puesto que son consideradas valiosas. En promedio responden de manera completa todas las secciones de la encuesta 120 participantes para una tasa de respuesta del 43%. Este comportamiento es similar al reportado en estudios previos en el área [PUN03] [GAR10] [GAR13] [GAR15].

Las respuestas que reportan mayor tasa de respuesta son las que pueden ser respondidas mediante un solo clic, por tanto, para futuros estudios se recomienda la categorización de posibles respuestas, basado en los resultados de este estudio para permitir a los participantes un menor tiempo de respuesta. De antemano conocíamos que el cuestionario podía considerarse largo para algunos participantes, pero dada la amplitud de temas analizados consideramos que se realizó el mejor trabajo posible para la configuración del mismo.

Al igual que lo reportado por [GAR15], confirmamos que la no adopción de las buenas prácticas en la ingeniería del *software* no solo se debe a aspectos técnicos. En muchos casos, la no adopción de estas prácticas está directamente relacionado con la falta de recursos en las organizaciones. Esto queda evidenciado por los desafíos planteados por los participantes donde solicitan a la comunidad académica realizar estudios para mejorar la productividad del desarrollo del software aplicando solo las prácticas esenciales para mantener la calidad del producto.

Se recomienda al lector de este reporte evitar atribuir la no adopción de las prácticas de la SE solo a aspectos técnicos, de conocimiento o destrezas de los profesionales. Este tipo de estudios, más bien, deben considerarse como material de referencia para encontrar oportunidades de mejora en la industria y consolidar nuestro ecosistema tecnológico.

### 6.3 Limitaciones y amenazas a la validez.

Las limitaciones en nuestro estudio son similares a las comúnmente reportadas en estudios previos en el área [PUN03] [GAR10] [GAR13] [GAR15]. A continuación se detallan las amenazas a la validez interna, constructo, conclusión y externa.

**Validez interna.** La validez interna es una propiedad de un estudio que refleja hasta qué punto la conclusión se basa en el contexto y los datos extraídos. Para este estudio, tenemos la amenaza del sesgo de la selección de los participantes puesto que es una muestra a conveniencia no aleatorizada. Nuestros participantes fueron invitados por correo electrónico por medio de las principales instituciones que tienen relación con el área de estudio. Se estima que el estudio debió llegar a la gran mayoría de profesionales relacionados con la ingeniería del software y el tiempo que la encuesta estuvo en línea lo consideramos suficiente para el llenado del instrumento. Se enviaron recordatorios periódicos a los participantes. Consideramos que obtuvimos una muestra representativa de la industria de Costa Rica, sin embargo consideremos que la participación del sector privado, principalmente pequeñas organizaciones y organizaciones transnacionales pudo ser mejor. Esto representa un desafío para futuras encuestas en el área. También consideramos que la participación de ingenieros jóvenes debe mejorar para una próxima encuesta.

**Validez del constructo.** La validez del constructo se refiere hasta qué punto el objeto de estudio representa las teorías estudiadas. Para este estudio, utilizamos documentos reconocidos en el área de la SE y procesamos la información de acuerdo a las prácticas comunes en este tipo de estudios. Para evitar que los participantes se sintieran evaluados se les indicó explícitamente que la participación en la encuesta era anónima y los resultados publicados serían mostrados de manera agregada sin afectar la privacidad de sus organizaciones.

**Validez de las conclusiones.** La validez de las conclusiones se refiere a que los resultados del estudio recojan las conclusiones a través de un proceso riguroso y replicable. En nuestro estudio, para cada pregunta de investigación se realizó un procesamiento de

datos estadístico y trazable a través de la información analizada. Los procedimientos son los mismos usados en estudios previos. Las preguntas abiertas son tabuladas si realizar ninguna interpretación de los datos y se muestran tal y como las detallaron los participantes.

**Validez externa.** La validez externa se refiere hasta qué punto se pueden generalizar los resultados del estudio. En nuestro estudio se aplicaron diferentes estrategias para minimizar el sesgo de la población. Sin embargo, los resultados de este estudio solo pueden generalizarse a los datos demográficos detallados en el estudio. Aunque conocemos por experiencias previas en la industria que muchos de nuestros resultados son comunes, no se pueden generalizar sin un análisis crítico en el cual los lectores deben evaluar estos resultados de acuerdo a diferentes contextos. Este estudio presenta muchas similitudes en comparación con estudios previos, lo que genera un aporte importante para la comunidad de la ingeniería del software.

## 7 Conclusiones y trabajo futuro

La industria de *software* en Costa Rica se ha consolidado en los últimos años, es una industria competitiva a nivel mundial y sigue en crecimiento constante. Nuestros profesionales son de alta calidad, y se interesan en aplicar las mejores prácticas disponibles y las nuevas metodologías y herramientas en sus procesos de desarrollo de *software*, si estas les generan valor agregado.

La encuesta revela información interesante sobre las diferentes tendencias en la industria de *software* en Costa Rica. Además, de los principales desafíos a los que se enfrenta la comunidad de la ingeniería de *software* del país, que incluye la academia y la industria privada y pública. La calidad de las soluciones tecnológicas y la productividad de los equipos de desarrollo son los dos principales desafíos a los que se enfrentan los profesionales. Debido a las restricciones de presupuesto, buscan determinar cuál es la mejor combinación de prácticas y actividades esenciales para mejorar la productividad manteniendo el nivel de calidad deseado.

De acuerdo a los resultados, la industria costarricense se encuentra en el momento justo para empezar a adoptar prácticas de aseguramiento de la calidad, mejoramiento y medición de procesos. Para todo esto, se deben realizar colaboraciones industria academia que generen beneficios para ambas partes. Esto debe incluir, no solo colaboraciones para probar nuevas metodologías o herramientas, sino también, las actividades de capacitación que permitan la adopción de estas nuevas tecnologías.

Las investigaciones no solo deben dirigirse a los aspectos técnicos de la SE, sino también, a los aspectos organizacionales y sociales de la SE. Finalmente, es necesario incentivar las colaboraciones industria y academia, que en muchos casos pueden aprovechar algunos de los incentivos que el gobierno pueda otorgar para tales efectos. Como parte de los resultados del estudio se han identificado gran cantidad de desafíos que pueden guiar estas futuras investigaciones.

Nuestro estudio confirma algunas de los resultados reportados en otros países, lo que consideramos un aporte importante para la comunidad de la ingeniería del *software*. Deben realizarse trabajos similares en distintos países, primero mapeando el estado general de adopción de las prácticas de la SE y luego analizando temas específicos de interés para la industria.

Cómo trabajo futuro, planeamos replicar el estudio para lograr un mayor alcance a la población de profesionales del país. Recomendamos que este estudio se pueda realizar de manera periódica para poder determinar el nivel de avance como industria en la adopción de las prácticas de la ingeniería de *software*. Además, debemos analizar como los factores demográficos influyen la adopción de cada una de las prácticas de la SE, este análisis de acuerdo a las recomendaciones de [GAR16]. Dicho análisis estudiará el impacto del perfil de los participantes, las organizaciones y los tipos de proyectos en la adopción de las prácticas de la SE. Finalmente, son necesarios estudios específicos en distintas áreas de conocimiento de la SE, para estudiar a un nivel de profundidad mayor, el estado de la industria en distintas áreas de interés.

**Agradecimientos.** Este estudio fue apoyado por el Ministerio de Ciencia, Tecnología y Telecomunicaciones (MICITT) y la Universidad de Costa Rica (No. 834-B5-A18). Agradecemos a todos(as) los(as) profesionales de la ingeniería del *software* que participaron anónimamente en la encuesta y a las organizaciones que colaboraron en la difusión del cuestionario: Universidades públicas y privadas, Instituciones del sector público, Cámara de Tecnologías de la Información y Comunicación (CAMTIC) y Colegio de Profesionales en Informática y Computación (CPIC).

## Referencias

[ADD02] Addison, T., & Vallabh, S. (2002, September). Controlling software project risks: an empirical study of methods used by experienced project managers. In Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology (pp. 128-140). South African Institute for Computer Scientists and Information Technologists.

[AYK09] Aykol, M. M. (2009). Software engineering and software management practices in Turkey. Bahçeşehir University, Turkey.

[AYT03] Aytac, T., Ikiz, S., & Aykol, M. (2003). A SPICE-oriented, SWEBOK-based software process assessment on a national scale: Turkish software sector survey. In International Conference on Process Improvement and Capability dEtermination (SPICE) Conference.

[BOE00] Boehm, B., Abts, C., & Chulani, S. (2000). Software development cost estimation approaches—A survey. *Annals of software engineering*, 10(1-4), 177-205.

[BON15] Bonilla, L. B. B., Baraya, A. R., Budden, M. C., & Bermudez, L. (2015). Revisiting The Software Industry In Costa Rica: Achievements And Challenges. *Journal of Service Science (Online)*, 8(1), 29.

- [BOU14] Bourque P., & Fairley, R., eds., (2014). *Guide to the Software Engineering Body of Knowledge, Version 3.0*, IEEE Computer Society; [www.swebok.org](http://www.swebok.org).
- [BOZ10] Bozkurt, M., Harman, M., & Hassoun, Y. (2010). Testing web services: A survey. Department of Computer Science, King's College London, Tech. Rep. TR-10-01.
- [BRE08] Brenes, L., & GoVaere, V. (2008). La industria del software en Costa Rica. En Comercio Exterior, 58(5).
- [CAM14] CAMTIC. (2014). Mapeo del Sector de las TICs de Costa Rica. Informe elaborado para las Cámara Costarricense de Tecnologías de Información y Comunicación (CAMTIC), bajo el patrocinio del Fondo Multilateral de Inversiones (FOMIN) del Banco Interamericano de Desarrollo (BID).
- [CAR10] Carver, J. C. (2010, May). Towards reporting guidelines for experimental replications: A proposal. In International Workshop on Replication in Empirical Software Engineering Research, Cape Town, South Africa.
- [CAU10] Causevic, A., Sundmark, D., & Punnekkat, S. (2010, April). An industrial survey on contemporary aspects of software testing. In 2010 Third International Conference on Software Testing, Verification and Validation (pp. 393-401). IEEE.
- [CHE07] Chen, W., Li, J., Ma, J., Conradi, R., Ji, J., & Liu, C. (2007, May). A survey of software development with open source components in Chinese software industry. In International Conference on Software Process (pp. 208-220). Springer Berlin Heidelberg.
- [CUR88] Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11), 1268-1287.
- [CUS03] Cusumano, M., MacCormack, A., Kemerer, C. F., & Crandall, W. (2003). A global survey of software development practices. Center for eBusiness@ MIT, Paper, 178, 1-17.
- [CUT03] Cuthbertson, C. (2003). The State of IT Project Management in the UK 2002-2003. Computer Weekly. Templeton College, University of Oxford.
- [DAP16] Dapozo, G., Medina, Y., & Lencina, B. (2016). La práctica de la estimación en empresas y áreas de Sistemas. *Revista Electrónica Argentina-Brasil de Tecnologías da Informação e da Comunicação*, 1(4).
- [DEA13] Deak, A., & Stålhane, T. (2013, March). Organization of testing activities in Norwegian Software Companies. In Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on (pp. 102-107). IEEE.
- [DEN07] Denger, C., Feldmann, R. L., Host, M., Lindholm, C., & Shull, F. (2007, September). A snapshot of the state of practice in software development for medical devices. In First International Symposium on Empirical Software Engineering and Measurement (ESEM 2007) (pp. 485-487). IEEE.

[DIA16] Dias-Neto, A. C., Matalonga, S., Solari, M., Robiolo, G., & Travassos, G. H. (2016). Toward the characterization of software testing practices in South America: looking at Brazil and Uruguay. *Software Quality Journal*, 1-39.

[DIP09] Canfora, G., & Di Penta, M. (2009). Service-oriented architectures testing: A survey. In *Software Engineering* (pp. 78-105). Springer Berlin Heidelberg.

[EGO09] Egorova, E., Torchiano, M., & Morisio, M. (2009, May). Evaluating the perceived effect of software engineering practices in the Italian industry. In *International Conference on Software Process* (pp. 100-111). Springer Berlin Heidelberg.

[EGO10] Egorova, E., Torchiano, M., & Morisio, M. (2010). Actual vs. perceived effect of software engineering practices in the Italian industry. *Journal of Systems and Software*, 83(10), 1907-1916.

[ENG10] Engström, E., & Runeson, P. (2010, June). A qualitative survey of regression testing practices. In *International Conference on Product Focused Software Process Improvement* (pp. 3-16). Springer Berlin Heidelberg.

[FUJ07] Fujita, H., & Pisanelli, D. (2007). *Software Engineering Practices: An Auditor's Perspective*. New Trends in Software Methodologies, Tools and Techniques, 161, 36.

[GAR10] Garousi, V., & Varma, T. (2010). A replicated survey of software testing practices in the Canadian province of Alberta: What has changed from 2004 to 2009?. *Journal of Systems and Software*, 83(11), 2251-2262.

[GAR13] Garousi, V., & Zhi, J. (2013). A survey of software testing practices in Canada. *Journal of Systems and Software*, 86(5), 1354-1376.

[GAR13] Garousi, V., Coşkunçay, A., Can, A. B., & Demirörs, O. (2013). A survey of software testing practices in Turkey. In *Turkish National Software Engineering Symposium (Ulusal Yazılım Mühendisliği Sempozyumu, UYMS)*.

[GAR15] Garousi, V., Coşkunçay, A., Betin-Can, A., & Demirörs, O. (2015). A survey of software engineering practices in Turkey. *Journal of Systems and Software*, 108, 148-177.

[GAR16] Garousi, V., Coşkunçay, A., Demirörs, O., & Yazici, A. (2016). Cross-factor analysis of software engineering practices versus practitioner demographics: An exploratory study in Turkey. *Journal of Systems and Software*, 111, 49-73.

[GER04] Geras, A. M., Smith, M. R., & Miller, J. (2004). A survey of software testing practices in Alberta. *Canadian Journal of Electrical and Computer Engineering*, 29(3), 183-191.

[GRI06] Grindal, M., Offutt, J., & Mellin, J. (2006, August). On the testing maturity of software producing organizations. In *Testing: Academic & Industrial Conference-Practice And Research Techniques (TAIC PART'06)* (pp. 171-180). IEEE.

[GRO00] Groves, L., Nickson, R., Reeve, G., Reeves, S., & Utting, M. (2000). A survey of software development practices in the New Zealand software industry. In Software Engineering Conference, 2000. Proceedings. 2000 Australian (pp. 189-201). IEEE.

[GRO09] Groves, R. M., Fowler Jr, F. J., Couper, M. P., Lepkowski, J. M., Singer, E., & Tourangeau, R. (2009). Survey methodology (Vol. 561). John Wiley & Sons.

[HEA15] Heaton, D., & Carver, J. C. (2015). Claims about the use of software engineering practices in science: A systematic literature review. *Information and Software Technology*, 67, 207-219.

[HOA02] Hoare, T. (2002, August). Assertions in modern software engineering practice. In Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International (pp. 459-459). IEEE.

[JAL12] Jalali, S., & Wohlin, C. (2012). Global software engineering and agile practices: a systematic review. *Journal of software: Evolution and Process*, 24(6), 643-659.

[KAN13] Kanij, T., Merkel, R., & Grundy, J. (2013, May). Lessons learned from conducting industry surveys in software testing. In Proceedings of the 1st International Workshop on Conducting Empirical Studies in Industry (pp. 63-66). IEEE Press.

[KIR12] Kirk, D., & Tempero, E. (2012, December). Software development practices in New Zealand. In 2012 19th Asia-Pacific Software Engineering Conference (Vol. 1, pp. 386-395). IEEE.

[KIT02] Kitchenham, B. A., & Pfleeger, S. L. (2002). Principles of survey research part 2: designing a survey. *ACM SIGSOFT Software Engineering Notes*, 27(1), 18-20.

[LEE12] Lee, J., Kang, S., & Lee, D. (2012, September). A survey on software product line testing. In Proceedings of the 16th International Software Product Line Conference-Volume 1 (pp. 31-40). ACM.

[LIN15] Linaker, J., Sulaman, S. M., Høst, M., & De Melo, R. M. (2015). Guidelines for conducting surveys in software engineering. <https://lup.lub.lu.se/search/publication/5366801>.

[MAT01] Mata, F., & Jofré, A. (2001). Informe final de Estudio de Oferta y Demanda del Recurso Humano. Programa de Apoyo a la Competitividad del Sector de Desarrollo de Software de Costa Rica (BID-CAPROSOFT-PROCOMER-FUNCENAT), San José, Costa Rica, junio.

[MOK03] Molokken, K., & Jorgensen, M. (2003, September). A review of software surveys on software effort estimation. In Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on (pp. 223-230). IEEE.

[MOL04] Moløkken-Østfold, K., Jørgensen, M., Tanilkan, S. S., Gallis, H., Lien, A. C., & Hove, S. W. (2004, September). A survey on software estimation in the Norwegian industry. In Software Metrics, 2004. Proceedings. 10th International Symposium on (pp. 208-219). IEEE.

[MUR04] Ng, S. P., Murnane, T., Reed, K., Grant, D., & Chen, T. Y. (2004). A preliminary survey on software testing practices in Australia. In Software Engineering Conference, 2004. Proceedings. 2004 Australian (pp. 116-125). IEEE.

[NAN15] Nanthaamornphong, A., & Carver, J. C. (2015). Test-Driven Development in scientific software: a survey. *Software Quality Journal*, 1-30.

[NAS06] Nasir, M. (2006, June). A survey of software estimation techniques and project planning practices. In Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'06) (pp. 305-310). IEEE.

[NAS06] Nasir, M., & Ahmad, H. F. (2006, July). An Empirical Study to Investigate Software Estimation Trend in Organizations Targeting CMMI<sup>®</sup> SM. In 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR'06) (pp. 38-43). IEEE.

[PAE12] Páez, I. D., Anaya, R., & Travassos, G. H. (2012). Estado actual de la estimación de software en compañías colombianas que han adoptado CMMI.

[PAR08] Park, J., Ryu, H., Choi, H. J., & Ryu, D. K. (2008, February). A survey on software test maturity in Korean defense industry. In Proceedings of the 1st India software engineering conference (pp. 149-150). ACM.

[PFL01] Pfleeger, S. L., & Kitchenham, B. A. (2001). Principles of survey research: part 1: turning lemons into lemonade. *ACM SIGSOFT Software Engineering Notes*, 26(6), 16-18.

[PRO14] PROSIC. (2014). Informe Anual 2013 Hacia la Sociedad de la Información y el Conocimiento. Universidad de Costa Rica, 2014.

[PUN03] Punter, T., Ciolkowski, M., Freimut, B., & John, I. (2003, September). Conducting online surveys in software engineering. In Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on (pp. 80-88). IEEE.

[REH07] Rehman, J. U., Jabeen, F., Bertolino, A., & Polini, A. (2007). Testing software components for integration: a survey of issues and techniques. *Software Testing, Verification and Reliability*, 17(2), 95-133.

[RUN03] Runeson, P., Andersson, C., & Höst, M. (2003). Test processes in software product evolution—a qualitative survey on the state of practice. *Journal of software maintenance and evolution: Research and practice*, 15(1), 41-59.

[RUN06] Runeson, P. (2006). A survey of unit testing practices. *IEEE software*, 23(4), 22-29.

[SAE05] Sanz, L. F. (2005). Un sondeo sobre la práctica actual de pruebas de software en España. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 1(2), 50.

- [SAN12] Santos, G., Kalinowski, M., Rocha, A. R., Travassos, G. H., Weber, K. C., & Antonioni, J. A. (2012, September). MPS. BR program and MPS model: main results, benefits and beneficiaries of software process improvement in Brazil. In *Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the* (pp. 137-142). IEEE.
- [SCH14] Schmidt, C. T., Ganesha Venkatesha, S., & Heymann, J. (2014, May). Empirical insights into the perceived benefits of agile software engineering practices: A case study from SAP. In *Companion Proceedings of the 36th International Conference on Software Engineering* (pp. 84-92). ACM.
- [SHU08] Shull, F. J., Carver, J. C., Vegas, S., & Juristo, N. (2008). The role of replications in empirical software engineering. *Empirical Software Engineering*, 13(2), 211-218.
- [SIL16] Siles, I., González, J. E., & Marengo, A. M. (2016). ¿El “Silicon Valley latinoamericano”? la producción de tecnología de comunicación en Costa Rica (1950-2016). *Anuario de Estudios Centroamericanos*, 42(1), 411-441.
- [SIN10] Singer, J., Lethbridge, T., Vinson, N., & Anquetil, N. (2010, November). An examination of software engineering work practices. In *CASCON First Decade High Impact Papers* (pp. 174-188). IBM Corp..
- [SJO05] Sjøberg, D. I., Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N. K., & Rekdal, A. C. (2005). A survey of controlled experiments in software engineering. *IEEE transactions on software engineering*, 31(9), 733-753.
- [SOM97] Sommerville, I., & Sawyer, P. (1997). Viewpoints: principles, problems and a practical approach to requirements engineering. *Annals of Software Engineering*, 3(1), 101-130.
- [SWE04] Guide to the Software Engineering Body of Knowledge 2004 Version, in *SWEBOK 2004 Guide to the Software Engineering Body of Knowledge*, 2004. URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4425813&isnumber=4425812>.
- [TAH10] Tahir, A., & Ahmad, R. (2010, December). Requirement engineering practices-an empirical study. In *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on* (pp. 1-5). IEEE.
- [TAI05] Taipale, O., Smolander, K., & Kälviäinen, H. (2005, November). Finding and ranking research directions for software testing. In *European Conference on Software Process Improvement* (pp. 39-48). Springer Berlin Heidelberg.
- [TAI06] Taipale, O., Smolander, K., & Kälviäinen, H. (2006). A survey on software testing. 6th International SPICE.
- [TEV04] Tevanlinna, A., Taina, J., & Kauppinen, R. (2004). Product family testing: a survey. *ACM SIGSOFT Software Engineering Notes*, 29(2), 12-12.
- [TOR03] Torkar, R., & Mankefors, S. (2003, November). A survey on testing and reuse. In *Software: Science, Technology and Engineering, 2003. SwSTE'03. Proceedings. IEEE International Conference on* (pp. 164-173). IEEE.

[TRA11] Travassos, G. H., & Kalinowski, M. Performance Results of Software Organizations that Adopted the MPS Model from 2008 to 2011.

[TRE08] Trendowicz, A., Münch, J., & Jeffery, R. (2008, October). State of the practice in software effort estimation: a survey and literature review. In IFIP Central and East European Conference on Software Engineering Techniques (pp. 232-245). Springer Berlin Heidelberg.

[VON12] Vonken, F., Brunekreef, J., Zaidman, A. E., & Peeters, F. (2012). Software Engineering in the Netherlands: The State of the Practice. Delft University of Technology, Software Engineering Research Group.

[YAN08] Yang, D., Wang, Q., Li, M., Yang, Y., Ye, K., & Du, J. (2008, October). A survey on software cost estimation in the chinese software industry. In Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement (pp. 253-262). ACM.

[ZAI08] Zainol, A. (2008, December). Investigation into requirements management practices in the Malaysian software industry. In Computer science and software engineering, 2008 international conference on (Vol. 2, pp. 292-295). IEEE.

[ZAI09] Zainol, A., & Mansoor, S. (2009). A Survey of Software Engineering Practice in the Software Industry. In Proceedings of IASTED International Conference on Software Engineering. Innsbruck, Austria.