

ANEXOS

Anexo A: Evaluación de la calidad de los estudios seleccionados.

Código	1	2	3	4	5	6	7	8	9	10	Total
	Estudio	Objetivos	Contexto	Diseño de experimento	Muestreo	Grupo de control	Análisis de datos	Sesgo	Resultados	Valor	
E01	1	1	1	1	1	0	1	0	1	1	9
E02	1	1	1	1	0	0	1	0	1	1	8
E03	1	1	1	1	0	0	1	0	1	1	8
E04	1	1	1	1	1	1	1	0	1	1	10
E05	1	1	1	1	1	1	1	0	0	1	9
E06	1	1	1	1	0	1	1	0	1	1	9
E07	1	1	1	0	0	1	1	0	1	1	7
E08	1	1	1	1	0	0	1	0	1	1	8
E09	1	1	1	1	0	0	1	0	1	1	8
E10	1	1	1	1	0	0	1	1	1	1	8
E11	1	1	1	0	0	0	1	0	1	1	7
E12	1	1	1	1	0	1	1	0	1	1	9
E13	1	1	1	1	0	0	1	0	1	1	8
E14	1	1	1	1	0	0	1	0	1	1	8
E15	1	1	1	1	1	1	1	0	1	1	10
E16	1	1	1	0	1	1	1	0	1	1	9
E17	1	1	1	0	0	0	1	0	1	1	7
E18	1	1	1	0	1	1	1	0	1	1	9
E19	1	1	1	1	1	1	1	0	1	1	10
E20	1	1	1	1	0	0	1	1	1	1	9
E21	1	1	1	1	0	0	1	0	1	1	8
E22	1	1	1	0	0	1	1	0	1	1	8
E23	1	1	1	1	0	0	1	0	1	1	8
E24	1	1	1	0	0	0	1	0	1	1	7
E25	1	1	1	0	0	1	1	0	1	1	8
E26	1	1	1	0	0	0	1	0	1	1	7
E27	1	1	1	1	1	1	1	0	1	1	10
E28	1	1	1	1	1	1	1	0	1	1	10
E29	1	1	1	0	0	0	1	0	1	1	7
E30	1	1	1	0	0	1	1	0	1	1	8
E31	1	1	1	0	0	0	1	0	1	1	7
E32	1	1	1	1	1	1	1	0	1	1	10
E33	1	1	1	1	1	1	1	0	1	1	10
E34	1	1	1	1	1	1	1	0	1	1	10
E35	1	1	1	1	1	1	1	0	1	1	10

E36	1	1	1	0	0	1	1	0	1	1	8
E37	1	1	1	0	0	1	1	0	1	1	8
E38	1	1	1	0	0	0	1	0	1	1	7
E39	1	1	1	1	0	1	1	0	1	1	9
E40	1	0	1	1	0	1	1	0	1	1	8
E41	1	1	1	1	0	1	1	0	1	1	9
E42	1	1	1	1	0	0	0	1	1	1	7
E43	1	1	1	1	1	0	1	0	1	1	9
E44	1	1	1	1	0	1	1	0	1	1	9
E45	1	1	1	1	0	0	1	0	1	1	8
E46	1	1	1	1	0	0	1	1	1	1	9
E47	1	1	1	1	0	0	1	0	1	1	8
E48	1	1	1	1	1	1	1	0	1	1	10
E49	1	1	1	1	0	0	1	1	1	1	9
E50	1	1	1	1	0	0	1	1	1	1	9
E51	1	1	1	1	0	0	1	1	1	1	9
E52	1	1	1	1	1	0	1	0	1	1	9
E53	1	1	1	1	1	0	1	0	1	1	9
E54	1	1	1	1	1	1	1	1	1	1	11
E55	1	1	1	1	1	0	1	1	1	1	10
E56	1	1	1	1	1	1	1	0	1	1	10
E57	1	1	1	1	1	0	1	1	1	1	10
E58	1	1	1	1	1	1	1	0	1	1	10
E59	1	1	1	1	0	0	1	1	1	1	9
E60	1	1	1	1	0	0	1	0	1	1	8
E61	1	1	1	1	0	0	1	1	1	1	9

Anexo B: Resumen de los estudios primarios seleccionados.

Cód	Título	Autor	Año	Tipo de Estudio	Práctica Ágil	Entorno	Resultados
E01	The Case for Collaborative Programming	J. T. Nosek	1998	Experimento	PP	Profesional	<ul style="list-style-type: none"> - Pares producen soluciones de mejor calidad (de legibilidad y funcionalidad). - Aumento de la moral (evaluación cualitativa).
E02	Strengthening the Case for Pair Programming	L. Williams et al.	2000	Experimento	PP	Académico	<ul style="list-style-type: none"> - Aumento de la calidad (número de casos de prueba aprobados). - (40 - 50%) menos tiempo que individualmente, pero a mayor costo.
E03	Integrating Unit Testing Into A Software Development Team's Process	R. Ynchausti	2001	Estudio de caso	TDD + TFD	Profesional	<ul style="list-style-type: none"> - Mejoras en la calidad variaron de 38% a 267% menos defectos.
E04	Experimental Evaluation of Pair Programming	J. Nawrocki and A. Wojciechowski	2001	Experimento	PP	Académico	<ul style="list-style-type: none"> - No hay diferencia en calidad (líneas de código y número de reenvíos debido a errores). - Ninguna diferencia en el tiempo de desarrollo.
E05	Experiment about Test-First programming	M.M. Müller, and O. Hagner	2002	Experimento	TFD	Académico	<ul style="list-style-type: none"> - No hay diferencias en la calidad. - Una mejor reutilización al usar TDD.
E06	The Effects of Pair-Programming on Performance in an Introductory Programming Course	C. McDowell et al.	2002	Experimento	PP	Académico	<ul style="list-style-type: none"> - Aumento de la calidad (funcionalidad y legibilidad). - Mejora la información y la transferencia de conocimiento.
E07	Towards empirical evaluation of TDD in a university environment	M. Pancur et al.	2003	Experimento	TDD	Académico	<ul style="list-style-type: none"> - Pequeña diferencia en la calidad externa (pruebas externas: ok). - Menor cobertura de código.
E08	Test-Driven Development as a Defect-Reduction Practice	L. Williams et al.	2003	Estudio de caso	TDD	Profesional	<ul style="list-style-type: none"> - Reducción de la tasa de defectos en un 40%.
E09	Assessing test-driven development at IBM	E.M. Maximilien, and L. Williams	2003	Estudio de caso	TDD	Profesional	<ul style="list-style-type: none"> - 50% menor tasa de defectos. No hay disminución de la productividad.
E10	Using Test Driven Development in the Classroom	S. H. Edwards	2003	Experimento	TDD	Académico	<ul style="list-style-type: none"> - 45% menos defectos al usar TDD.
E11	When Does a Pair Outperform Two Individuals?	K.M. Lui and K.C.C. Chan	2003	Experimento	PP	Profesional	<ul style="list-style-type: none"> - Aumento de la calidad y productividad. - Mejor y más rápido diseño de algoritmos.
E12	Pair programming and pair trading: effects on learning and motivation in a CS2 course	T.H. DeClue	2003	Experimento	PP	Académico	<ul style="list-style-type: none"> - Aumento de la calidad (diseño/código). - Mejora de la calidad del trabajo en equipo, habilidades de comunicación, comprensión y aprendizaje.
E13	Experiences in learning xp practices	B. Tessem	2003	Estudio de caso	XP + PP + TDD	Académico	<ul style="list-style-type: none"> - PP lleva a una mejora en la calidad. - TFD produce código más alta calidad/diseño. - Mejor estimación de trabajo al usar planning game.
E14	A formal experiment comparing extreme programming with traditional construction	F. Macias et al.	2003	Experimento	XP	Académico	<ul style="list-style-type: none"> - No hay diferencia en la calidad, ya sea interna o externa entre los equipos XP y los tradicionales. - No hay diferencia en el tamaño del producto entre los equipos XP y los tradicionales.
E15	A structured experiment of test-driven development	B. George and L. Williams	2004	Experimento	TDD	Profesional	<ul style="list-style-type: none"> - Los programadores TDD producen código de mejor calidad, ya que pasaron 18% mas casos de prueba

							funcionales de caja negra.
E16	Program quality with pair programming in CS1	B. Hanks et al.	2004	Experimento	PP	Académico	<ul style="list-style-type: none"> - Estudiantes en pares producen programas más cortos y menos complejos. - Más confianza en su trabajo.
E17	Exploring extreme programming in context	L. Layman et al.	2004	Estudio de caso	XP	Profesional	<ul style="list-style-type: none"> - Mejora del 65% en la calidad preliminar, y del 35% en la calidad posterior al lanzamiento. - Aumento del 46% en la productividad.
E18	Analyses of an agile methodology implementation	S. Ilieva et al.	2004	Estudio de caso	XP	Profesional	<ul style="list-style-type: none"> - 13% menos defectos reportados. - Aumento del 42% en la productividad.
E19	A cross-program investigation of student's perceptions of agile methods	G. Melnik, and F. Maurer	2005	Mixto	XP	Académico	<ul style="list-style-type: none"> - 76% sugirió que XP mejora la calidad del código. - 78% cree que el uso de XP mejora la productividad en los equipos pequeños.
E20	On the effectiveness of the test first approach to programming	H. Erdogmus et al.	2005	Experimento	TFD	Académico	<ul style="list-style-type: none"> - La calidad externa mínima aumentó con el número de pruebas. - Los estudiantes TF escribieron más pruebas. - Más resultados consistentes de calidad al usar TDD.
E21	A Multiple Case Study on the Impact of Pair Programming on Product Quality	H. Hulkko, and P. Abrahamsson	2005	Estudio de caso	PP	Profesional	<ul style="list-style-type: none"> - Contraste en los resultados de la densidad de defectos (calidad). - Contraste en los resultados de la productividad.
E22	Two controlled experiments concerning the comparison of pair programming to peer review	M.M. Müller	2005	Experimento	PP	Académico	<ul style="list-style-type: none"> - Para un mismo nivel de corrección, la programación en pares y en solitario tienen el mismo costo. - Para diferentes niveles de corrección los pares producen con mejor calidad a expensas de un incremento en los costos.
E23	Comparison of student experiences with plan-driven and agile methodologies	C.A. Wellington et al.	2005	Experimento	XP	Académico	<ul style="list-style-type: none"> - Mediciones de calidad mejoradas. - Significativamente mayor calidad del código.
E24	A cross-program investigation of students' perceptions of agile	G. Melnik and F. Maurer	2005	Cualitativo Encuesta	TFD	Académico	<ul style="list-style-type: none"> - 73% de los estudiantes percibe que TFD mejora la calidad
E25	Evaluating the efficacy of TDD: industrial case	T. Bhat, and N. Nagappan	2006	Estudio de caso	TDD	Profesional	<ul style="list-style-type: none"> - Mayor tiempo de desarrollo 15% (proyecto B), 35% (proyecto A). - Disminución de defectos/KLOC en un 62% (proyecto A), 76% (proyecto B)
E26	Results from introducing component level test automation and TDD	L.O. Damm and L. Lundberg	2006	Estudio de caso	TDD	Profesional	<ul style="list-style-type: none"> - 5-30% de disminución en la tasa de errores iniciales. - Disminución del 60% en el costo de fallas evitables. - Coste total del proyecto se redujo en 6.5%.
E27	Pair programming improves student retention, confidence, and quality	C. McDowell, et al.	2006	Estudio de caso	PP	Académico	<ul style="list-style-type: none"> - Estudiantes en pares producen programas de mejor calidad, tienen más confianza, y lo disfrutaron más. - Programadores mujeres se benefician de la programación en pares.
E28	Critical Personality Traits in Successful Pair Programming	J. Chao and G. Atli	2006	Encuesta Experimento	PP	Profesional Académico	<ul style="list-style-type: none"> - Ciertos rasgos de personalidad aseguran código de mejor calidad (mente abierta y responsabilidad). - Diferencias estadísticamente significativas.
E29	Is External Code Quality Correlated with Programming Experience or	L. Madeyski	2006	Experimento	PP	Académico	<ul style="list-style-type: none"> - La calidad del externa (número de pruebas de aceptación pasadas) se correlaciona con el factor de sentirse bien, y el uso de

	Feelgood						pruebas clásicas.
E30	Pair programming productivity: novice–novice vs. expert–expert	K.M. Lui and K.C.C. Chan	2006	Experimento	PP	Académico	<ul style="list-style-type: none"> - La programación en parejas ayuda a resolver con eficacia los problemas de programación poco familiares. - Pares de novatos contra novatos individualmente son mucho más productivos que pares de expertos contra expertos individuales.
E31	Empirical Validation of Test-Driven Pair Programming in Game Development	S. Xu and V. Rajlich	2006	Estudio de caso	PP + TDD + Refac	Académico	<ul style="list-style-type: none"> - Mejor calidad (mayor número de casos de prueba, código más limpio, con mayor cohesión, cantidad de métodos más razonable).
E32	On the Sustained Use of a TDD Practice at IBM	C. Sanchez et al.	2007	Estudio de caso	TDD	Profesional	<ul style="list-style-type: none"> - Mejora de la calidad externa e interna para los mismos productos. - El uso de TDD disminuye el grado en el que aumenta la complejidad del código.
E33	An experimental evaluation of the effectiveness and efficiency of the TDD	A. Gupta and P. Jalote	2007	Experimento	TDD	Académico	<ul style="list-style-type: none"> - Mejora la calidad externa del código (afectado por los esfuerzos de las pruebas). - Reduce los esfuerzos globales de desarrollo. - Mejora la productividad de los desarrolladores.
E34	Quality impact of introducing component level test automation and TDD	L.O. Damm and L. Lundberg	2007	Estudio de caso	TDD	Profesional	<ul style="list-style-type: none"> - La proporción de errores se redujo de 60-70% a 0-20%
E35	Evaluating Pair Programming with Respect to Complexity and Expertise	E. Arisholm et al.	2007	Experimento	PP	Profesional	<ul style="list-style-type: none"> - En el sistema más complejo, los programadores por pares tenían un aumento del 48% en la proporción de soluciones correctas. - No hay diferencias significativas en el tiempo necesario para resolver las tareas correctamente.
E36	Evaluating performances of pair designing in industry	G. Canfora et al.	2007	Experimento	PP	Profesional	<ul style="list-style-type: none"> - Mejora de la calidad es superior al 15%. - Incremento de esfuerzo para completar tareas.
E37	Do programmer pairs make different mistakes than solo programmers?	M.M. Müller	2007	Experimento	PP	Académico	<ul style="list-style-type: none"> - La programación en pares se adecua a problemas complejos y desafiantes. - Menos defectos de expresión en pares.
E38	Realizing quality improvement through TDD	N. Nagappan et al.	2008	Estudio de caso	TDD	Profesional	<ul style="list-style-type: none"> - La densidad de defectos de los cuatro productos se redujo entre un 40% y un 90% comparado a los proyectos similares que no utilizaron la práctica de TDD. - Aumento de 15-35% en el tiempo de desarrollo inicial después de la adopción de TDD.
E39	Does TDD Really Improve Software Design Quality?	D. Janzen and H. Saiedian	2008	Experimento Estudio de caso	TDD	Profesional Académico	<ul style="list-style-type: none"> - Los programadores TFD tienden a escribir software en pequeñas unidades menos complejas y altamente probadas. - El análisis de acoplamiento no proporciona respuestas claras. - La cohesión no se mejora.
E40	Pair programming: what's in it for me?	A. Begel and N. Nagappan	2008	Cualitativo Encuesta	PP	Profesional	<ul style="list-style-type: none"> - 65,4% de la programación en pares produce código de mejor calidad. - Menos errores, código comprensible. - Aumento del tiempo de trabajo, manejo de problemas y conflictos de personalidad.
E41	Pair programming in software development	T. Bippet al.	2008	Estudio de caso	PP	Académico	<ul style="list-style-type: none"> - Menor tiempo que en la programación en solitario. - Mayor conocimiento del código y mejora en su calidad.

	teams						- Pérdida de mínima de eficiencia.
E42	A Case Study on the Impact of Refactoring on Quality and Productivity in an Agile Team	R. Moser et al.	2008	Estudio de caso	XP + Refac	Profesional	- Refactoring conduce a una mayor calidad y productividad. - Mejores resultados para equipos pequeños.
E43	Does Test-Driven Development Improve the Program Code? Alarming Results from a Comparative Case Study	M. Siniaalto and P. Abrahamsson	2008	Estudio de caso	TDD	Académico/Profesional	- TDD produce código menos complejo (calidad externa) - Sin embargo, la estructura de paquetes se hace más difícil de cambiar y mantener.
E44	Empirical investigation towards the effectiveness of TFD	L.Huang and M. Holcombe	2009	Experimento	TFD	Académico	- La calidad externa de las aplicaciones entregadas aumentó con el porcentaje de tiempo dedicado a las pruebas, independientemente de la estrategia de pruebas. - Mayor esfuerzo en las pruebas. - 70% más de productividad, pero la mejora no es estadísticamente significativa.
E45	Implications of integrating TDD into CS1/CS2 curricula	C. Desai, and D. S. Janzen	2009	Experimento	TDD	Académico	- Ensayos del grupo TL tuvieron 39% más defectos que su contraparte TF (calidad externa). - No hay un aumento de la calidad interna.
E46	An Experimental Investigation of Personality Types Impact in Pair Programming	P. Sfetsos et al.	2009	Experimento	PP	Académico	- Grupos heterogéneos de personalidad son más eficientes (más pruebas de aceptación, mejor diseño/codificación, mayor velocidad y mejor comunicación).
E47	A quantitative approach for evaluating the effectiveness of refactoring in software development process	K. Usha et al.	2009	Estudio de caso	Refact	Académico	- Mejora en la reutilización y la comprensibilidad del código, mejorando así la capacidad de mantenimiento.
E48	Distributed agile: project management in a global environment	S. Lee and H.S Yong	2010	Estudio de caso	Scrum	Profesional	- Los proyectos ágiles distribuidos y Scrum muestran un 30% de mejora en la calidad. - Mayor satisfacción del cliente.
E49	Evaluating the Effect of Agile Methods on Software Defect Data and Defect Reporting	K. Korhonen	2010	Estudio de caso	TDD + PP + Scrum	Profesional	- El número de defectos se redujo y se mejoró el tiempo de solución de errores con el pasar del tiempo.
E50	The impact of test-first programming on branch coverage and mutation score indicator of unit tests	L. Madeyski	2010	Experimento	TFP	Académico	- Los beneficios del uso de TFD en este contexto específico pueden considerarse de importancia mínima.
E51	A comparative Analysis of the Agile and Traditional Software Development Processes Productivity	W. Carvalho, et al.	2011	Estudio de caso	Scrum hybrid	Profesional	- Incremento de la productividad en un 16% en los proyectos híbridos de Scrum y tradicional vs los RUP. - Resultados aplicados a proyectos medianos y pequeños solamente.
E52	Impact of test-driven development on productivity, code and tests	M. Pančur and M. Ciglarič	2011	Experimento	TDD + PP	Académico	- No se encontraron diferencias significativas en cuanto a calidad externa, productividad y pruebas de aceptación entre TDD y el modelo tradicional.

E53	Scrum + engineering practices: Experiences of three microsoft teams	L. Williams, et al.	2011	Estudio de caso	Scrum	Profesional	<ul style="list-style-type: none"> - Hasta un 250% de mejora en la productividad (lineas de codigo producidas al 4to sprint). - Menor densidad de defectos (0.53 por linea de codigo) y disminución del esfuerzo.
E54	The effectiveness of test-driven development: an industrial case study	T. Dogša and D. Batič	2011	Estudio de caso	TDD	Profesional	<ul style="list-style-type: none"> - TDD developers produced higher quality code that is easier to maintain - It is observed a reduction in productivity
E55	A Case Study Analyzing the Impact of Software Process Adoption on Software Quality	R. Tufail, and A. A. Malik	2012	Estudio de caso	Scrum	Profesional	<ul style="list-style-type: none"> - Scrum mejora la calidad mediante la reducción de los errores y defectos de alta gravedad (50%). - Aumento de la relación de pruebas pasadas/falladas (casi 7 veces más).
E56	Development of Auxiliary Functions: Should You Be Agile?	O. A. Lemos, et al.	2012	Experimento	TDD + PP	Académico	<ul style="list-style-type: none"> - PP tiende a aumentar la fiabilidad en términos de correctitud (menor número de errores). - TDD tiende a aumentar la fiabilidad en términos de tamaño de pruebas y cobertura. - En general se muestra un impacto negativo en el esfuerzo.
E57	Quality of Testing in Test Driven Development	A. Cauevic et al.	2012	Experimento	TDD	Académico	<ul style="list-style-type: none"> - Diferencia entre TDD y tradicional es casi inexistente. Sin embargo 29% de los casos de prueba fueron casos "negativos" (basado en requisitos no especificados). - El código creado por TDD es de mejor calidad en comparación al tradicional, sin embargo ambos presentan aun muchos errores.
E58	Refactoring with Unit Testing: A Match Made in Heaven?	F. Vonken and A.Zaidman	2012	Experimento		Académico	<ul style="list-style-type: none"> - Las pruebas unitarias disponibles durante la refactorización no conduce necesariamente a mejorar la velocidad en la refactorización. - Las pruebas unitarias durante la refactorización no aumenta la calidad del código de resultante.
E59	A Replicated Experiment on the Effectiveness of Test-first Development	D. Fucci and T. Burak	2013	Experimento	TFD	Académico	<ul style="list-style-type: none"> - No se encontró ninguna diferencia significativa en términos de productividad y de calidad externa. - TFD no mejora ni empeora la calidad externa ni la productividad
E60	An experimental evaluation of TDD vs. test-last development with industry professionals	H. Munir, et al.	2014	Experimento	TDD	Profesional	<ul style="list-style-type: none"> - No se muestra ninguna diferencia estadísticamente significativa a favor de TDD. - Potencial de reducir el número de defectos sin embargo, se necesitan más datos para estar seguros.
E61	Systematic analyses and comparison of development performance and product quality of Incremental Process and Agile Process	A. Tarhan and S. G. Yilmaz	2014	Estudio de caso	XP + Scrum	Profesional	<ul style="list-style-type: none"> - Rendimiento de desarrollo y calidad del producto superior: productividad (79%), densidad de defectos (57%), relación de esfuerzo de resolución de errores (26%), eficacia (21%), y predicción de esfuerzo (4%).