

EXPERIMENTACIÓN EN INGENIERÍA DE SOFTWARE: CASO DE ESTUDIO PROGRAMACIÓN EN PAREJA

Experimentation in Software Engineering: Pair Programming Case Study

José Julián Ventura Uscanga	zs22017046@estudiantes.uv.mx
Patricia Martínez Moreno	pmartinez@uv.mx
Luis Antonio Morales Velázquez	luismorales04@uv.mx
José Antonio Vergara Camacho	jvergara@uv.mx

¹ Facultad de Contaduría y Administración, Universidad Veracruzana, Coatzacoalcos, Veracruz, México.

RESUMEN

Debido a los beneficios de la programación en pareja en el desarrollo de software como el aumento en la producción de código de calidad y la transferencia de conocimientos entre un equipo de desarrollo, esta práctica ha captado el interés de educadores en Ingeniería de Software como una estrategia efectiva para desarrollar competencias clave en los estudiantes. Con el objetivo de evaluar su efectividad como método de enseñanza, se realizó un estudio con estudiantes de Ingeniería de Software, aplicando actividades relacionadas con la programación. Al comparar los resultados en términos de calificaciones y calidad del código, tanto individual como en pareja, y tras aplicar pruebas estadísticas, se concluyó que la programación en pareja mejora el desempeño general de los alumnos. Los resultados demostraron que las calificaciones y la calidad del código en pareja fueron superiores en promedio, confirmando que esta práctica es un método de enseñanza efectivo y beneficioso.

Palabras Clave: Programación en pareja, programación, calidad de código

ABSTRACT

Due to the benefits of pair programming in software development, such as increased production of quality code and knowledge transfer within a development team, this practice has captured the interest of educators in Software Engineering as an effective strategy for developing key skills in students. To evaluate its effectiveness as a teaching method, a study was conducted with Software Engineering students, applying programming-related activities. When comparing the results in terms of grades and code quality, both individually and in pairs, and after applying statistical tests, it was concluded that pair programming improves students' overall performance. The results showed that grades and code quality in pairs were higher on average, confirming that this practice is an effective and beneficial teaching method.

Keywords: Pair programming, programming, code quality

► I. Introducción

La programación individual es típicamente la primera forma en la que un programador aprende y se familiariza con todo lo que conlleva la práctica, y a medida que avanza, desarrolla programas cada vez más complejos aplicando su propia lógica y creando su propio estilo de programación. Por otra parte, un programador puede estar al tanto o no de la calidad de su código. La calidad del código implica muchos aspectos entre los que destacan los principios de un código limpio, su eficiencia y optimización. En el ámbito profesional, el código fuente define gran parte de calidad de un sistema [1] por lo que se han creado estilos de programación que mejoran el proceso y aseguran la calidad, como la programación en pareja.

La programación en pareja es una práctica popularizada con la creciente tendencia de las metodologías ágiles en los proyectos de software. Existen testigos de esta práctica desde 1953 [2], sin embargo, es XP la que define a la programación en pareja como una práctica primaria para la construcción de software en un proyecto [3]. En los últimos años también se ha evaluado el desempeño de esta práctica, tanto en el ámbito profesional como en el educativo de la programación debido a sus beneficios; ha demostrado influir en aspectos como la calidad del código y la rapidez de su producción, el estado de ánimo de los programadores, la comunicación efectiva entre equipos de desarrollo, el aprendizaje entre pares y la transferencia de conocimientos [2]

Específicamente en la región de Coatzacoalcos de la Universidad Veracruzana en la que toma lugar el presente estudio, la Ingeniería de Software es un área con oportunidades escasas, por lo que también presenta una disminución en la presencia de empresas u organizaciones especializadas, las cuales contratan programadores capaces para la construcción de sistemas. Por ello, los educadores de Ingeniería de Software de la zona deben encontrar métodos de enseñanza efectivos para el desarrollo de las habilidades de la programación de los estudiantes, donde la programación en pareja puede tratarse de una estrategia aplicable de manera que logre una mejor retención y aplicación de los conocimientos revisados en clase, además

de promover el interés y la estancia de los alumnos en sus respectivos cursos.

Programación en pareja. La programación en pareja es un estilo de programación en la que dos programadores trabajan lado a lado en una computadora, colaborando continuamente en el mismo diseño, algoritmo, código o prueba [2]. Una persona juega el rol de conductor (driver), que es la que escribe o realiza acciones en la computadora; y otra hace de navegador (navigator). El navegador es el responsable de identificar los errores tácticos y estratégicos del conductor [2]. Los errores tácticos son, por ejemplo, los errores de sintaxis, ortográficos o llamar a la función equivocada. Los errores estratégicos son, por ejemplo, cuando el conductor está codificando una función que produce cierto resultado en un programa, pero este resultado está alejado a lo que realmente debería cumplir la función. Es deber del navegador identificar este tipo de situaciones, comunicarle al conductor sus observaciones y discutir una solución de manera conjunta.

Muchos de los beneficios de la programación en pareja son documentados por Williams y Kessler [2]. En su obra, mencionan que la programación en pareja influye positivamente en la calidad del código y la rapidez de su producción. El código es producido con menos defectos y además en la mitad del tiempo que toma en comparación con la programación individual. Este estilo también construye la confianza y mejora el trabajo en equipo. Por último, destacan el aprendizaje mejorado. Las parejas adquieren nuevos conocimientos según observan a su compañero trabajar, como su habilidad de resolución de problemas, el cómo utilizan las capacidades de un lenguaje de programación, o cómo emplean las herramientas de desarrollo.

► II. Trabajos relacionados

Por la manera en que la programación en pareja puede beneficiar a estudiantes de la programación, ha habido un gran enfoque dentro del campo de la educación en Ciencias de la Computación sobre cómo y por qué la programación en pareja es una técnica exitosa [4] y efectiva para el

aprendizaje y el aumento del desempeño en sus cursos. Por ello, este estilo de programación ha sido ampliamente estudiado y aplicado en cursos reales de programación de diferentes instituciones educativas para su evaluar si su implementación funciona como un método de enseñanza valioso.

En un estudio [5] realizado en la Universidad de Tecnología y Negocios de Chongqing, China en 2010 se hizo un experimento con 59 estudiantes que formaban parte de un curso de programación en Java. Los estudiantes fueron emparejados con base en habilidades de programación similares. El estudio fue llevado a cabo para evaluar el impacto de la programación en pareja en la calidad del trabajo de los estudiantes, así como la evolución de la confianza de sus habilidades de programación. Al terminar el experimento, un 70% de los estudiantes acordaron que la confianza en sus habilidades mejoró gracias a la programación en pareja. Las calificaciones obtenidas en pareja fueron más altas que los individuales, concluyendo que los artefactos producidos por las parejas son de mayor calidad. Por último, los estudiantes comentaron positivamente acerca del aprendizaje mejorado y de la calidad trabajando en pares.

En 2013 se realizó un estudio [4] en el Politécnico de Otago, Nueva Zelanda en el curso de Programación I donde se aplicaron actividades individuales y en pareja a 40 estudiantes. Las parejas fueron asignadas de forma que el nivel de habilidad (1, 2 y 3) de ambos estudiantes era similar. El experimento examinó la evolución de la confianza de las habilidades aplicando la programación en pareja y su efectividad como un método de enseñanza efectivo. Los resultados mostraron que los estudiantes con habilidades de nivel 1 completaron un 84% más actividades en pareja. Además, la proporción de estudiantes de nivel 1 disminuyó alrededor de un 22%, y obtuvieron calificaciones más altas. Finalmente, los estudiantes comentaron positivamente acerca de la programación en pareja destacando la retroalimentación constante por sus parejas, la resolución de problemas mejorada, y una socialización más amplia con sus compañeros. Los docentes concluyeron que la programación en pareja sirve como un método de enseñanza efectivo

y beneficioso para los estudiantes principiantes de la programación, sugiriendo la aplicación de este estilo en cursos similares.

» III. Diseño del experimento

Con el fin de determinar que la programación en pareja es una forma de enseñanza valiosa que puede mejorar el aprendizaje, el rendimiento académico y la calidad del trabajo de los estudiantes de la licenciatura en Ingeniería de Software, se llevó a cabo un experimento en la Universidad Veracruzana Campus Coatzacoalcos con el objetivo de evaluar la implementación de la programación en pareja en un curso de quinto periodo con un total de 15 estudiantes en la asignatura de Paradigmas de Programación, durante la introducción al paradigma de POO.

3.1 Selección de sujetos

Para el experimento, participaron 15 estudiantes de quinto periodo de la Licenciatura en Ingeniería de Software. Los estudiantes contaban con experiencia programando, habiendo completado los cursos de Introducción a la Programación, Programación, Estructuras de Datos y Principios de Construcción de Software. Los conocimientos de los estudiantes se tomaron como suficientes para que pudieran llevar a cabo las actividades diseñadas con normalidad. Por otra parte, el tamaño de la muestra fue considerado como limitante para el experimento, y se tomaron medidas para contrarrestar la falta de fiabilidad de los resultados, como el análisis de la normalidad de los datos y la selección de una prueba de contraste de hipótesis adecuada para el análisis de los resultados.

3.2 Selección de variables

Variables independientes. Las variables independientes identificadas para este experimento son las respectivas estrategias o estilos de programación aplicados por actividad, es decir, la programación individual y en pareja.

Variables dependientes. Se identificaron dos variables dependientes que tienen como objetivo

medir cuantitativamente el desempeño de los estudiantes, siendo estas la calificación y el puntaje de calidad del código de sus entregables.

3.3 Instrumentación

Para recopilar los datos, se diseñaron dos actividades diferentes relacionadas con el paradigma de la POO. La primera consistía en realizar un ejercicio de manera individual aplicando este paradigma donde se debía crear una aplicación que controlara un registro de vehículos. Para la segunda tarea se definió que los estudiantes formaran parejas libremente según su preferencia. El ejercicio consistió en un crear una aplicación similar a la anterior en nivel de complejidad, con la diferencia de que se trataba de la gestión de una biblioteca. Las actividades ponían en práctica conceptos tales como las clases y sus relaciones, herencia y encapsulación.

Se emplearon dos rúbricas para evaluar el trabajo de los alumnos. La primera rúbrica trata sobre los criterios para la asignación de calificaciones, donde la puntuación máxima es de 100 puntos. La segunda rúbrica se utilizó para evaluar los siguientes aspectos de la calidad del código: Dominio del lenguaje Java, Fundamentos de POO, Claridad y Mantenibilidad, Uso de Patrones de Diseño, Documentación y Comentarios, Optimización de rendimiento, Manejo de recursos, Resolución de problemas y Depuración, que sumaba un total de 36 puntos.

3.4 Ejecución del experimento

Antes de aplicar las actividades a los estudiantes, se les comentó brevemente que participarán en un experimento para evaluar la programación en pareja, y se explicó cuál es la dinámica de dicha práctica. Las actividades se llevaron en dos sesiones diferentes en un laboratorio de cómputo, en el cual se aseguró que los alumnos se distribuyeran a lo largo de laboratorio para facilitar la programación en pareja. También, se les comentó que mantuvieran buenas prácticas de codificación y documentación a lo largo de su código. Al finalizar cada actividad, los estudiantes debían entregar sus evidencias a través de un repositorio en GitHub.

3.5 Validación de datos

Los datos obtenidos fueron los puntajes asignados de calificación y calidad del código tras la evaluación del trabajo de los estudiantes de forma individual y en pareja utilizando las rúbricas preestablecidas. Todos los estudiantes entregaron satisfactoriamente sus evidencias. Se detectaron dos valores atípicos en la calificación y el puntaje de calidad del código correspondiente a los mismos sujetos al aplicar la primera actividad en individual. Estos valores modificaron notablemente la distribución de los datos, debido principalmente al tamaño de la muestra. También se omitió el resultado de un estudiante en pareja para mantener la homogeneidad de dicho tamaño

» IV. Análisis e interpretación de los resultados

El procedimiento de análisis está principalmente basado en la metodología de proceso experimental propuesto por Piattini, M [6]. A continuación, se muestran los resultados obtenidos del experimento en las dos actividades aplicadas para evaluar la programación individual y en pareja. La tabla I muestra la calificación y el puntaje de la calidad del código de los estudiantes en dichas actividades. La Fig. 1 muestra los diagramas de cajas y bigotes con el fin de demostrar la dispersión de los datos obtenidos e identificar diferencias entre los grupos.

Tabla I
PUNTAJES OBTENIDOS POR ESTUDIANTE

Estudiante	Calificación		Calidad del código	
	Individual	Parejas	Individual	Parejas
Estudiante 1	65	90	32	33
Estudiante 2	90	95	31	32
Estudiante 3	35	70	10	24
Estudiante 4	85	90	18	32
Estudiante 5	45	70	14	24
Estudiante 6	90	90	30	33
Estudiante 7	85	90	34	32
Estudiante 8	90	95	32	34
Estudiante 9	90	90	34	32
Estudiante 10	85	90	30	33
Estudiante 11	95	95	31	33
Estudiante 12	75	90	24	33
Estudiante 13	70	90	24	32
Estudiante 14	35	90	15	33
Estudiante 15	85	95	29	34

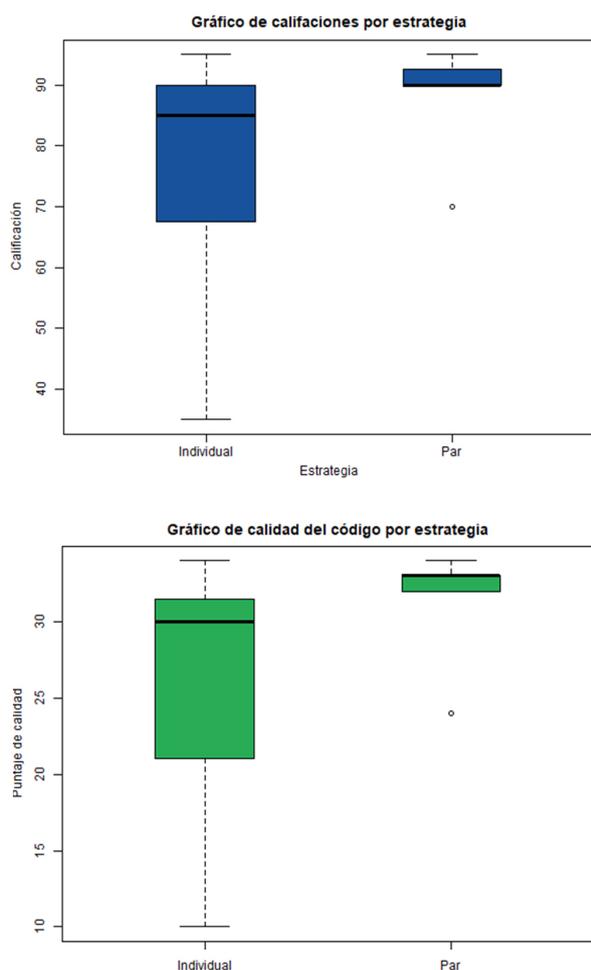


Fig. 1. Diagramas de calificaciones y calidad del código por estrategia.

A primera instancia, se puede deducir que el rendimiento de los alumnos mejoró aplicando la programación en pareja. De la Fig.1, se puede observar que los diagramas poseen una distribución similar, de la cual se puede destacar que la estrategia individual posee una dispersión y varianza de datos más notable que en pareja.

4.1 Estadística descriptiva

En la tabla II se muestran el promedio, la mediana y la desviación estándar de las calificaciones y la puntuación de calidad del código obtenidas por los estudiantes trabajando de forma individual y en pareja. Se observa que la calificación y la calidad del código obtenidos en pareja es mayor que en individual. La desviación estándar también indica que hubo una variación de datos mayor en la estrategia individual, mientras que, en pareja,

las calificaciones y el puntaje de la calidad son más consistentes respecto a la media.

Tabla II
ESTADÍSTICA DESCRIPTIVA DE CALIFICACIONES Y CALIDAD DEL CÓDIGO.

Origen	N	Calificación			Calidad del código		
		\bar{x}	Mediana	s	\bar{x}	Mediana	s
Individual	15	74.666	85	20.569	25.866	30	7.945
Parejas	15	88.666	90	7.898	31.6	33	3.157

4.2 Análisis de normalidad

Primeramente, se determinó si se posee una distribución normal de la diferencia de las calificaciones y el puntaje de la calidad del código entre las estrategias aplicadas a través de la prueba de Shapiro-Wilk, con el fin de seleccionar qué tipo de prueba se aplicará sobre los datos. En la tabla III se resumen los resultados de la prueba de normalidad.

Tabla III
RESULTADOS DE LA PRUEBA DE SHAPIRO-WILK PARA CALIFICACIONES Y CALIDAD DEL CÓDIGO.

Origen	Prueba de Shapiro-Wilk			
	α	W	p	Rechazar H_0
Calificaciones individual	0.05	0.794	0.003	si $0.003 > 0.05$
Calificaciones pareja	0.05	0.623	$4.201e-05$	si $4.201e-05 > 0.05$
Calidad individual	0.05	0.852	0.018	si $0.018 > 0.05$
Calidad pareja	0.05	0.601	$2.624e-05$	si $2.624e-05 > 0.05$

Estableciendo un valor de $\alpha = 0.05$, en la prueba de hipótesis sobre la normalidad para el conjunto de datos correspondiente a las calificaciones y calidad del código individual y en parejas se rechaza la hipótesis nula, indicando que no cuentan con una distribución normal. Conociendo las normalidades de los datos, es posible establecer que para la prueba de contraste de hipótesis se ocupará la prueba no paramétrica de rangos con signos Wilcoxon.

4.3 Comparación de resultados

Para establecer que existe una diferencia entre el desempeño de los estudiantes de forma individual

y en pareja, se aplicó la prueba no paramétrica para muestras dependientes de rangos con signo de Wilcoxon de cola superior. Se optó por emplear una prueba para muestras dependientes, debido a que se tratan de los mismos sujetos experimentales evaluados y la finalidad es comparar su desempeño antes y después de aplicar la programación en pareja. Por otra parte, las condiciones del experimento son casi idénticas con la excepción que, en la primera actividad, los estudiantes trabajaban individualmente; los criterios de evaluación no se modificaron y la temática de la actividad continuaba sobre el paradigma de POO. A continuación, se describe la interpretación de la prueba de hipótesis de dicha prueba:

Prueba de rangos con signos de Wilcoxon de cola superior. La prueba de hipótesis para el rechazo de la hipótesis nula es la siguiente:

- H0: La diferencia de las medianas de los puntajes entre parejas e individual es menor o igual que 0.

$$M_d \leq 0 \tag{1}$$

- H1: La diferencia de las medianas de los puntajes entre parejas e individual es mayor que 0.

$$M_d > 0 \tag{2}$$

donde $M_d = (M_{parejas} - M_{individual})$.

Los siguientes apartados contienen la interpretación de las pruebas de hipótesis para los dos criterios evaluados:

Calificaciones. En la tabla IV se resumen los resultados de la prueba de Wilcoxon. Considerando un nivel de significancia $\alpha = 0.05$ y el valor de $p = 0.001$, es posible rechazar la hipótesis nula, por lo que la diferencia de las medianas de las calificaciones entre la estrategia en pareja e individual es mayor que 0. Con ello es posible concluir que la mediana de las calificaciones en pareja es mayor que en individual, demostrando que los estudiantes consiguieron mejores calificaciones trabajando en pareja.

Tabla IV
RESULTADOS DE PRUEBA DE WILCOXON PARA CALIFICACIONES

Calificaciones	Prueba de Wilcoxon			
	α	V	p	Rechazar H_0
Resultados	0.05	0	0.001	Si $0.001 < 0.05$

Calidad del código. La tabla V resume los resultados obtenidos de la prueba de Wilcoxon. Considerando un nivel de significancia $\alpha = 0.05$ y el valor de $p = 0.002$ obtenido, se rechaza la hipótesis nula, por lo que la diferencia de las medianas de la calidad del código entre la estrategia en pareja e individual es mayor que 0. Con ello es posible concluir que la mediana de la calidad del código en pareja es mayor que en individual, demostrando que el código de los estudiantes mostró una mejor calidad en pareja.

Tabla V
RESULTADOS DE PRUEBA DE WILCOXON PARA CALIDAD DEL CÓDIGO

Calidad del código	Prueba de Wilcoxon			
	α	V	p	Rechazar H_0
Resultados	0.05	9	0.002	Si $0.002 < 0.05$

4.4 Amenazas a la validez

Validez interna. Se consideraron los siguientes aspectos como amenazas a la validez interna: el emparejamiento de los estudiantes, la experiencia adquirida, y los sujetos al tanto del experimento. El primero hace referencia a la manera en que fueron emparejados los estudiantes; para el experimento, se les indicó que formaran parejas de manera libre, lo que pudo influenciar en los resultados obtenidos ya sea positiva o negativamente.

La experiencia adquirida es en cuestión de la adquisición de conocimientos para resolver la segunda actividad; al completar la primera tarea, los alumnos aprendieron cómo solucionar los problemas dados, entonces, es lógico que para la segunda actividad sabrán cómo resolverlo de manera más rápida y sencilla. El último aspecto es sobre la información dada del experimento a los estudiantes; esta consideración es importante, ya que el desempeño de los sujetos puede ser fácilmente manipulado, de forma que para la primera actividad pudieron no haberse esforzado en completarla, y para la segunda, realizarla de la

mejor manera, alterando la validez de la conclusión del experimento.

Validez externa. La única amenaza detectada para la validez externa fue la selección de sujetos, donde en el presente experimento se consideró como adecuado, ya que el contexto del estudio y los sujetos están relacionados y al final fueron los ideales para llevar a cabo el experimento.

Validez de la conclusión. Entre las amenazas identificadas se tomaron en cuenta el tamaño de la muestra y el poder estadístico de la prueba de contraste de hipótesis seleccionada. El tamaño de la muestra fue considerado como limitante para el experimento debido a que era bajo y fue determinante para la normalidad de los datos. Debido al tamaño de la muestra y la normalidad, se aplicó una prueba no paramétrica para muestras pareadas, la prueba de rangos con signos de Wilcoxon, la cual posee un poder estadístico relativamente inferior a comparación de su contraparte paramétrica, la prueba T pareada.

» V. Conclusiones

Tras la aplicación del experimento, la recopilación de resultados y realización del análisis estadístico empleando las pruebas adecuadas al tamaño de muestra limitada para determinar las pruebas de hipótesis sobre la comparación de la programación en pareja se determinó que esta última se trata de una estrategia aplicable en la enseñanza de la Ingeniería de Software que tiene un impacto positivo en el rendimiento académico y de la calidad del código en los estudiantes de Ingeniería de Software, de tal manera que el promedio de calificaciones obtenidas aumentaron y el código producido cumplió más con los criterios de calidad establecidos, además de representar puntajes más consistentes y con menor variación. Si bien es cierto, que es un experimento ya probado y comprobado como se muestra en el apartado trabajos relacionados, cabe señalar que, para este contexto y región geográfica no se había realizado.

Los resultados concuerdan con los estudios relacionados con la experimentación de la programación en pareja, en los cuales se logró

demostrar que esta estrategia mejora el desempeño de los estudiantes y la calidad de su trabajo. Dichos datos sugieren que la programación en pareja es un método de enseñanza efectivo y beneficioso, y que su implementación cumple con el objetivo de lograr un mayor desarrollo de las habilidades y conocimientos de los alumnos.

» VI. Referencias

- [1] McConnell, S. Code Complete. A practical handbook of software construction. Microsoft Press, 2004.
- [2] Williams, L. y Kessler, R. Pair Programming Illuminated. Addison-Wesley, 2003.
- [3] Beck, K. y Andres, C. Extreme programming explained. Second Edition. Addison-Wesley, 2004.
- [4] Wood, K., Parsons, D., Gasson, J., y Haden, P. "It's never too early: pair programming in CS1". In Proceedings of the Fifteenth Australasian Computing Education Conference - Volume 136 (ACE '13). Australian Computer Society, Inc., AUS, 2013, pp. 13–21. *Biol.* 147, 195 197 (1981). doi: 10.1016/0022-2836(81)90087-5
- [5] Lai, H. y Xin, W. "Experimental research of the pair programming in java programming course," Proceeding of the International Conference on e-Education, Entertainment and e-Management, Bali, Indonesia, 2011, pp. 257-260.
- [6] Piattini, M. Métodos de investigación en ingeniería del software. Ra-Ma, 2014.