

## Apropiación de prácticas del proceso de software personal: un estudio de caso

### Appropriating personal Software Process practices: A case study

Óscar Mauricio Granada<sup>1</sup>  
Sergio Augusto Cardona Torres<sup>2</sup>  
Sonia Jaramillo Valbuena<sup>3</sup>

Recibido: 10/04/2017 - Aceptado: 28/07/2017

Cómo citar este artículo: [#] O. Granada, S. Cardona y S. Jaramillo “Apropiación de prácticas del proceso de software personal: un estudio de caso”, *IngEam*, vol. 4, n.º 1 pp. 71- 86, 2017

#### Resumen

El proceso de software personal (PSP) fue concebido para aplicar buenas prácticas individuales de desarrollo de software. Este trabajo tuvo como propósito implementar una estrategia de formación, para identificar la apropiación de un subconjunto de buenas prácticas de desarrollo de software personal, en curso avanzado de programación de computadores. Los estudiantes presentaron buenos resultados en los conceptos relacionados con calidad en el software, el concepto de métrica, el registro de defectos y el diligenciamiento de resumen de plan de proyectos.

**Palabras clave:** desarrollo de software, procesos de software personal, métricas de calidad.

#### Abstract

The Personal Software Process (PSP) was designed to apply good individual software development practices. The purpose of this work was to implement a training strategy, to identify the appropriation of a subset of good personal software development practices, in advanced computer programming course. The students presented good results in the concepts related to software quality, the concept of metrics, the Registry of defects and the summary diligence of Project plan.

**Key words:** Software development, personal software processes, quality metrics.

---

<sup>1</sup> Ingeniero de Sistemas y Computación - Universidad del Quindío. Correo electrónico: omauriciog@gmail.com

<sup>2</sup> Doctor en Ingeniería - Universidad del Quindío. Correo electrónico: sergio\_cardona@uniquindio.edu.co

<sup>3</sup> Doctor en Ingeniería - Universidad del Quindío. Correo electrónico: sjaramillo@uniquindio.edu.co

## 1. Introducción

La dinámica actual de las organizaciones requiere de soluciones de software que puedan soportar la complejidad de su operación y garanticen estar alineadas con sus objetivos estratégicos, por esta razón las empresas de desarrollo de software deben construir productos basados en procesos formales y disciplinados a partir de estándares internacionales de calidad. Es importante que los países emergentes aprovechen el potencial que pueden tener en el sector de la construcción de software, fortaleciéndolo en las capacidades de los equipos de trabajos y las personas. El recurso humano en las empresas de TI y específicamente el programador de software, es considerado un recurso crítico e indispensable para alcanzar el éxito de los proyectos de desarrollo.

El desarrollo de software es una actividad que requiere de recurso humano que reúna cada vez más competencias y habilidades con el fin de satisfacer con las necesidades y requerimientos de las aplicaciones de software de la actualidad. Estas habilidades no sólo se componen de aspectos técnicos como lógica para la solución de problemas o la sintaxis de los lenguajes de programación; sino que también es necesario poseer habilidades blandas, que permitan hacer una administración del trabajo por parte del mismo programador. Conscientes de la importancia del recurso humano en la industria del software, se propone el proceso personal de desarrollo de software (PSP) [1], el cual pretende contribuir a la aplicación de buenas prácticas al momento de desarrollar software, por parte de los programadores. Así mismo, PSP promueve una mejora en el trabajo de los programadores que aplican estas prácticas en sus procesos de construcción de software, permitiéndoles estimar y planificar sus actividades, controlar su rendimiento según su planeación y mejorar la calidad de sus programas [2].

El desarrollo de estas habilidades en etapas tempranas de la formación de los estudiantes, permite crear una cultura de disciplina personal y laboral a la hora de construir aplicaciones de software, facilitar su incorporación al medio laboral y ejecutar un trabajo más predecible y posiblemente menos riesgoso. La instrucción en PSP se convierte en un valor agregado que las personas que desarrollan software aportan en sus proyectos y en la industria del desarrollo de aplicaciones que se encuentra ávida de productos y servicios con calidad.

Con base en lo anterior, es necesario que, desde las instituciones de educación superior, se promuevan prácticas de desarrollo de software de acuerdo a estándares internacionales. Sin embargo, en el ámbito académico son pocas las iniciativas relacionadas con la adopción de modelos o metodologías que mejoren sus habilidades en estimación, planeación, calidad en su trabajo, cumplimiento de cronograma y reducción de defectos en sus productos [3]. La ausencia de estas habilidades en la formación de los programadores, puede generar la poca obtención de capacidades que les permita hacer un trabajo más predecible y con mayor posibilidad de cumplimiento en el alcance y tiempo de sus asignaciones.

Es común que en los programas académicos relacionados con la informática, se centren en la adquisición de habilidades y capacidades técnicas para la construcción de software [4]. Se empiezan a desarrollar programas sin aplicar principios básicos de análisis, solución de problemas y de gestión de sus actividades, y sin realizar una reflexión acerca de la calidad de sus productos. Es por ello que el concepto de

calidad no se encuentra debidamente caracterizado y por lo tanto no se posee una cultura de aplicación de buenas prácticas en el proceso de desarrollo de software. La formación de habilidades para desarrollar software con calidad es impartida finalizando los programas de pregrado o en cursos especializados de formación para egresados. Para ese momento, el programador ya posee un proceso de construcción de software definido y puede presentar dificultades en la asimilación e incorporación de nuevas actividades en su proceso. Es importante que esas competencias sean adquiridas por los estudiantes desde el inicio de su formación profesional y en lo posible, en paralelo con la adquisición de las habilidades técnicas [2].

Teniendo en cuenta que particularmente PSP implica un método riguroso para la recopilación y el análisis de información, estas acciones no son un proceso sencillo y se corre el riesgo que el estudiante no lo perciba como un elemento que le agregue valor a su formación, viéndolo como un método que no le aporta a su proceso de desarrollo de software personal. En muchas ocasiones las actividades propuestas por PSP pueden ser entendidas como un llenado de formatos, que resultan poco motivadores e interesantes para los estudiantes [5].

El objetivo de la investigación es mostrar los resultados de la implementación de una estrategia de formación, para identificar la apropiación de un subconjunto de buenas prácticas de desarrollo de software personal, con estudiantes de un curso avanzado de programación de computadores.

Este artículo en su estructura, se presenta de la siguiente manera, en la sección 2 se presenta el marco de referencia a partir del cual se evidencia el aporte de la investigación. En la sección 3 se presenta la metodología empleada para el desarrollo de la presente investigación. En la sección 4 se presenta el análisis de los resultados obtenidos. Finalmente, se presentan las conclusiones del trabajo.

## **2. Marco de referencia**

Son diversos los reportes de experiencias académicas relacionadas con la apropiación de prácticas de PSP. Se han identificado diferentes investigaciones relacionadas con el impacto de PSP sobre la formación de los estudiantes [6]. Se han realizado estudios empíricos en cursos de programación a nivel de pregrado [7], y a nivel de posgrado [8]. El PSP también ha sido utilizado para experimentar en cursos de Ingeniería de Software [9], [10]. Como propósito común de ellas, se busca que los estudiantes en etapa de formación adopten este modelo internacional de calidad en la construcción de software, y en las cuales se han registrado inconvenientes como el descrito por [3] en su investigación en la calidad del entorno mexicano en el desarrollo de software, al concluir que se requiere de gran cantidad de tiempo por parte de los docentes que imparten los cursos relacionados con las prácticas PSP, para dar retroalimentación inmediata de los trabajos y ejercicios de los estudiantes, hacerles un acompañamiento permanente e impartir los temas y conceptos del proceso.

En [11], se realizó un trabajo orientado a motivar a los estudiantes de un curso de ingeniería de software para que vieran el desarrollo de software como una disciplina sistemática antes que como una actividad que se genera a partir de ensayo y error, encontraron dificultades al tratar de mantener a los estudiantes enfocados en las ideas generales del proceso, generar en ellos la suficiente auto-disciplina, obtener una

herramienta de soporte adecuada y en resaltar los diferentes tipos de registro que se realizan en PSP. Por su parte, [4] en la introducción de PSP0 en cursos de la Tecnología de Sistemas de la Corporación Universitaria Adventista en Medellín, Colombia, manifiesta dentro de sus resultados, encontrarse con dificultades de los docentes para realizar el monitoreo de los estudiantes, dificultad para dar una retroalimentación inmediata y relevante e inconvenientes al ajustar las asignaciones prácticas alineadas con las habilidades reales de los estudiantes. En [5], al realizar una prueba piloto de su estrategia de enseñanza de PSP, propuesta para un curso de programación de Ingeniería de Sistemas de la Universidad del Quindío, en Armenia, Colombia, se afirma que los estudiantes no evidenciaron mejoras en la estimación del tamaño del producto, la planeación de actividades, el trabajo en equipo y en la aplicación de estándares de codificación.

Con base en el referencial teórico analizado y las problemáticas allí descritas, se hizo necesario realizar el diseño de una estrategia de enseñanza que permitiera a los estudiantes apropiarse las competencias y habilidades, y les encaminara a emplear buenas prácticas en su proceso personal de desarrollo de software, con la calidad como una característica implícita en sus labores como programadores. De manera adicional, en las experiencias académicas anteriormente mencionadas, se evidencia un objetivo común, orientar a los estudiantes en las buenas prácticas de desarrollo que recomienda PSP, sin embargo, carecen de una retroalimentación y métodos de medición orientadas a conocer el nivel de obtención de los logros y la asimilación de las prácticas planteadas por parte de los estudiantes.

### 3. Estrategia de formación

La estructuración del proceso de aprendizaje del curso se hizo a partir de la metodología de proyectos formativos, esto considerando que el trabajo práctico de los estudiantes de programación es bastante similar al trabajo real para el cual son formados.

La definición de la ruta formativa permitió articular los elementos que la componen: Presentación del proyecto, las competencias a desarrollar, la estructura formal del programa, las actividades del proyecto y la estructura de la evaluación. Las competencias están divididas en dos grupos, el primero tiene que ver con las habilidades técnicas que los estudiantes deben desarrollar y que están relacionadas en la carta descriptiva del curso de Programación Avanzada I. La otra parte son las competencias en las prácticas PSP que se desea que los estudiantes puedan apropiarse.

La tabla 1 muestra las competencias que se planearon desarrollar en los estudiantes.

Tabla 1. Competencias del proyecto formativo.

<p>Competencias técnicas</p>	<p>Permitir al estudiante hacer uso de herramientas técnicas para solucionar aspectos de concurrencia en las soluciones.                      Aprender a desacoplar con éxito las capas de presentación y lógica de negocio.                      Estar en capacidad de trabajar con aplicaciones distribuidas y las diferentes formas de establecer conexiones a través de redes de computadores.                      Comprender y utilizar la API JPA de Java.</p>
------------------------------	---

Competencias PSP	Construir programas de tamaño mediano y pequeño ejecutando las fases relacionadas con PSP nivel 0. Aprender a gestionar la métrica del tiempo haciendo correcto uso del formato de registro de tiempos, registro de interrupciones, con datos completos y consistentes. Registrar correctamente el formato de registro de defectos PSP. Entender el estándar de defectos PSP que permita tenerlo como base para el registro de defectos. Entender y seguir los guiones de PSP nivel 0. Entender el uso y la importancia de los estándares de programación al construir software. Aplicar el formato de registro de pruebas en la fase correspondiente. Usar de manera consistente el formato de mejora de proceso durante la etapa de Post-Mortem.
---------------------	---

La intervención tuvo una duración de dos meses, tiempo durante el cual se desarrollaron 16 sesiones de trabajo con los estudiantes. La valoración de los diferentes entregables que los estudiantes generaron, se realizó a partir de rúbricas, instrumentos que permitieron agrupar los criterios a evaluar y los niveles que definían la evaluación de las competencias relacionadas con la apropiación de las buenas prácticas para el proceso personal de desarrollo de software PSP.

#### 4. Metodología

El diseño de la investigación fue de corte cuasi experimental, con diseño intergrupos (control y experimental), con medida pretest – postest. Basado en un enfoque cuantitativo que posibilitó predecir las variables que fueron objeto de estudio. El diseño permitió validar la estrategia de formación para incorporar buenas prácticas de desarrollo PSP. El análisis descriptivo, se basó en la observación a los diferentes entregables realizados por los estudiantes y el análisis de su proceso personal de desarrollo de software evidenciado; e inferencial para analizar cómo la formación en PSP apoyada en una estrategia de enseñanza, puede influir en la apropiación de buenas prácticas de programación por partes de los estudiantes.

Se aplicó un instrumento de pretest, para conocer el nivel inicial de uso de las buenas prácticas de PSP de los estudiantes. Posteriormente, se realizó el diseño de la estrategia de enseñanza que orientó el aprendizaje de los estudiantes y la evaluación de las competencias PSP adquiridas. La estrategia de enseñanza se construyó a partir de la metodología de proyectos formativos [12], para el desarrollo de competencias. La estrategia diseñada se puso en práctica con el grupo experimental, el cual, de manera adicional, recibió la retroalimentación respectiva, conforme a sus avances en las actividades, mientras que el grupo control estuvo orientado bajo la estrategia tradicional de enseñanza. El mismo instrumento que sirvió para la obtención de información inicial, se usó para realizar el postest con ambos grupos, permitiendo conocer el impacto que la estrategia de enseñanza tuvo en la formación del grupo experimental y realizando la comparación con los avances que en ese sentido pudo tener el grupo control.

#### 4.1. Muestra

La población para la realización del estudio está compuesta por los estudiantes inscritos en los cursos de Programación Avanzada I de quinto semestre y Programación Avanzada II con estudiantes de sexto semestre del pregrado, del programa de Ingeniería de Software de la Escuela de Administración y Mercadotecnia del Quindío. La participación de los estudiantes fue de forma voluntaria mediante consentimiento informado. Se recalcó que el análisis se haría respetando el carácter confidencial de los datos y de acuerdo al decreto 1377 de 2013 [13], del régimen general para la protección de datos personales en Colombia.

#### 4.2. Instrumento de recolección de datos

Los instrumentos de recolección de la información fueron:

- Encuesta de uso de buenas prácticas de desarrollo de software (instrumento de pretest y postest): Encuesta con preguntas agrupadas de acuerdo a las etapas: planeación, gestión de tiempo, defectos, y proceso, para diagnosticar las diferentes prácticas de desarrollo de software que los estudiantes tienen al momento de iniciar y finalizar la intervención. El instrumento contó con 31 preguntas en total, agrupadas por categorías, como lo muestra la tabla 2. Se estableció una escala de Likert, de uno a cinco, en donde: (1) nunca lo realiza, (2) en muy pocas ocasiones (3) a veces (4) casi siempre y (5) siempre. Se realizó un proceso de validación con 6 desarrolladores de software, certificados en PSP Developer.

Tabla 2. Instrumento de recolección de información diagnóstica

Categoría	Cantidad de preguntas
Proceso	7
Tiempo	7
Tamaño	5
Defectos	8
Planeación	4

- Test de nivel de conocimiento de PSP: Test de conocimiento que se realizó al final del curso.

Durante la estrategia de formación, se usaron formatos basados en PSP, que permitieron registrar las métricas del proceso con el cual se construye software y posibilita el conocimiento de aspectos de su proceso personal de software.

## 5. Resultados

### 5.1. Diagnóstico sobre prácticas PSP de los estudiantes del grupo control y experimental

Para el diagnóstico de los estudiantes de los grupos control y experimental, se descartó un análisis de tipo paramétrico, dado que las variables que se manejaron en el diagnóstico son de tipo ordinal. Por esto, se seleccionó una técnica no paramétrica, como la prueba de Kruskal-Wallis, para realizar pruebas de hipótesis para cada una de las preguntas del instrumento. Esta prueba no paramétrica requiere el cumplimiento de los supuestos estadísticos de normalidad y de igualdad de varianzas, de los datos. La hipótesis que se plantea es:

- *H<sub>0</sub>*: “Existe diferencia estadísticamente significativa en aplicación de buenas prácticas de desarrollo, entre los estudiantes del grupo control y experimental, al inicio del proceso formativo”.

A continuación, se presenta la prueba de hipótesis para cada pregunta, de acuerdo a las categorías de PSP propuestas: proceso, tiempo, tamaño, defectos y planeación. En la tabla 3, se presenta el análisis de homogeneidad para la categoría proceso.

Tabla 3. Análisis de homogeneidad categoría proceso

Categoría	Pregunta	Estadístico de prueba	p-valor
Proceso	1	0,912371	0,339484
	2	0,0324074	0,857136
	3	0,849772	0,356615
	4	3,20689	0,0733249
	5	0,840841	0,359155
	6	1,09279	0,295851
	7	0,447443	0,503551

El p-valor para la categoría proceso, permite afirmar que no existe diferencia estadísticamente significativa en los estudiantes del grupo control y experimental, con relación a las preguntas relacionadas con la categoría proceso. A continuación, en la tabla 4, se presenta el análisis de homogeneidad para la categoría tiempo.

Tabla 4. Análisis de homogeneidad categoría tiempo

Categoría	Pregunta	Estadístico de prueba	p-valor
Tiempo	8	0,073347	0,786524
	9	0,051207	0,820976
	10	0,373707	0,54099
	11	0,82232	0,364501
	12	0,486111	0,485666
	13	0,112092	0,737775
	14	0,140292	0,707386

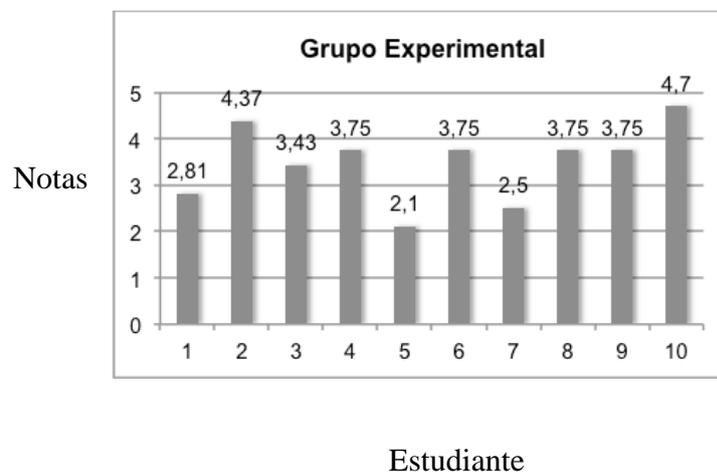
El p-valor para la categoría tiempo, permite afirmar que no existe diferencia estadísticamente significativa en los estudiantes del grupo control y experimental, con relación a las preguntas relacionadas con la categoría proceso.

La prueba de hipótesis, generó para todas las preguntas de las categorías: tamaño, defectos y planeación, un p-valor  $> 0.05$ , a partir de lo cual se puede afirmar que no existe diferencia estadísticamente significativa para estas categorías entre los grupos control y experimental, al inicio del proceso de formación, y por lo tanto se rechaza la hipótesis nula. Con lo anterior, se consideró factible implementar la estrategia de formación en el grupo experimental.

## 5.2. Prueba de conocimientos

Las preguntas fueron agrupadas con las mismas categorías PSP mencionadas anteriormente y cuyo propósito fue validar los conceptos de las diferentes prácticas. El examen tuvo 16 preguntas, todas con una igual ponderación, que sumadas generaron una nota entre 0 y 5. Todas las preguntas fueron de selección múltiple con única respuesta, agrupadas por las mismas categorías de la encuesta de buenas prácticas de programación, distribuidas así: 4 preguntas de proceso, 4 preguntas de administración de tiempo, 3 preguntas relacionadas con la métrica de tamaño, 3 preguntas de gestión de los defectos y 2 preguntas relacionadas con planeación. A continuación, en la figura 1, se presentan los resultados obtenidos por los estudiantes del grupo experimental.

Figura 1, resultados obtenidos por los estudiantes del grupo experimental.



Fuente: Elaboración propia

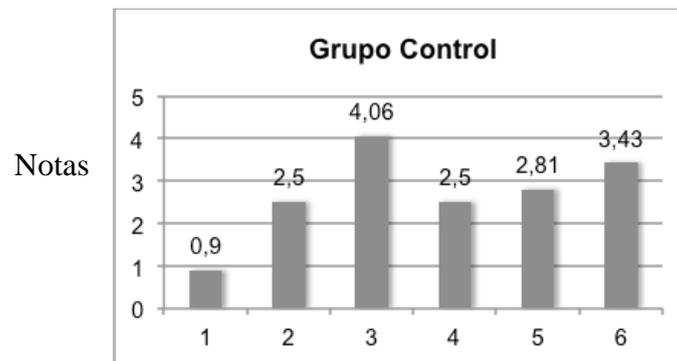
En la tabla 5 se presentan los resultados de cada una de las categorías de PSP, para el grupo experimental. Se evidencia para cada categoría que los estudiantes han apropiado adecuadamente los conceptos de cada una.

Tabla 5. Resultados por categoría de examen de conocimientos, grupo experimental

Categoría	Porcentaje bien respondidas	Porcentaje mal respondidas
Proceso	85%	15%
Tiempo	62,5%	37,5%
Tamaño	83,33%	16,67%
Defectos	43,33%	56,67%
Planeación	80%	20%

Por su parte, la figura 2 muestra las notas obtenidas por estudiantes del grupo control. Se evidencia una variabilidad muy alta con relación a los resultados. No todos los estudiantes han apropiado adecuadamente los conceptos de cada una de las buenas prácticas.

Figura 2. Notas de examen de conocimientos – Grupo control



Fuente: Elaboración propia

Estudiante

Al realizar la revisión de los exámenes se encuentra que el promedio del grupo experimental es de 3,49, mientras que el promedio del grupo control fue de 2,7. La tabla 6 muestra la cantidad de preguntas correctamente respondidas y erradas por cada categoría, en el grupo control.

Tabla 6. Resultados por categoría de examen de conocimientos, grupo control

Categoría	Promedio bien respondidas	Porcentaje mal respondidas
Proceso	62,50%	37,50%
Tiempo	45,83%	54,17%
Tamaño	66,67%	33,33%
Defectos	33,33%	66,67%
Planeación	66,67%	33,33%

Con base en los anteriores resultados, se pueden realizar las siguientes afirmaciones:

1. La categoría donde hay un mayor dominio por parte del grupo experimental es la de proceso, con un 85% de acierto. Para el grupo control las categorías de mayor dominio fueron tamaño y planeación, con un 66,67% de acierto.
2. La categoría donde más errores cometieron ambos grupos fue la de defectos con un 56,67% para el grupo experimental y 66,67% para el grupo control.
3. La brecha en conocimiento más grande entre el grupo experimental y el grupo control está en la categoría de proceso, donde hay un 22,5% de diferencia, a favor del grupo experimental.
4. En todas las categorías, el nivel de acierto del grupo control estuvo por debajo que el nivel de acierto del grupo experimental.
5. Dos estudiantes del grupo experimental tuvieron notas superiores a 4.0, mientras que un estudiante del grupo control superó ese valor.
6. En el grupo experimental el 70% de los estudiantes ganó el examen, mientras que en el grupo control sólo el 33% pudieron ganarlo.

### 3.3. Análisis de las prácticas PSP de los estudiantes del grupo control y experimental

Para el análisis de la apropiación de prácticas de PSP de los estudiantes, se hizo un estudio similar al del diagnóstico. Se aplicó la prueba no paramétrica de Kruskal-Wallis, mediante la cual se hizo una prueba de hipótesis para cada una de las preguntas del instrumento. La hipótesis que se plantea es:

- *Ho*: “Existe diferencia estadísticamente significativa en aplicación de buenas prácticas de desarrollo, entre los estudiantes del grupo control y experimental, al final del proceso formativo”.

Se presenta a continuación, la prueba de hipótesis para cada pregunta del instrumento, de acuerdo a las categorías de PSP propuestas: proceso, tiempo, tamaño, defectos y planeación. En la tabla 7, se presenta el análisis de homogeneidad para la categoría proceso, posterior a la implementación de la estrategia de formación

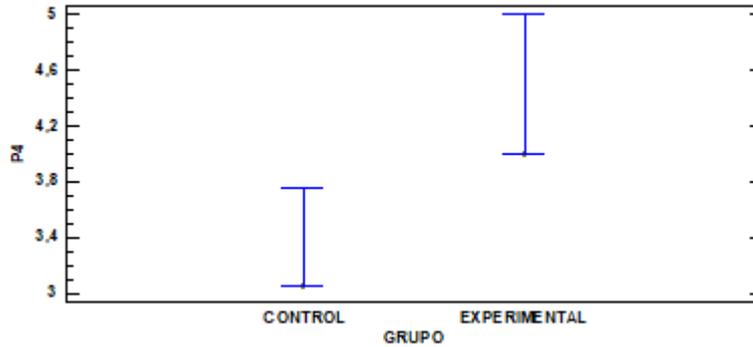
Tabla 7. Análisis de homogeneidad categoría proceso

Categoría	Pregunta	Estadístico de prueba	p-valor
Proceso	1	6,09026	0,0135906
	2	10,7768	0,00102726
	3	10,725	0,00105645
	4	9,17945	0,00244656
	5	12,5263	0,000400992
	6	9,51603	0,00203603
	7	5,48145	0,0192167

El p-valor para la categoría proceso, permite afirmar que existe diferencia estadísticamente significativa en los estudiantes del grupo control y experimental, con relación a las preguntas relacionadas con la categoría proceso. En la figura 3, el grafico de medianas muestra la diferencia estadística entre los estudiantes del grupo control y experimental, con relación a la pregunta 4 (p4): cuando construye sus

programas, el problema principal a resolver lo divide en problemas más pequeños para facilitar su solución.

Figura 3. Diagrama de medianas, pregunta 4



Fuente: Elaboración propia

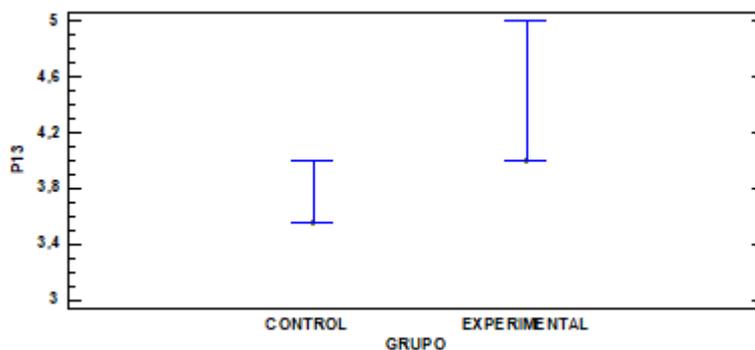
En la tabla 8, se presenta el análisis de homogeneidad para la categoría tiempo.

Tabla 8. Análisis de homogeneidad categoría tiempo

Categoría	Pregunta	Estadístico de prueba	p-valor
Tiempo	8	11,9306	0,000551843
	9	10,9302	0,000945605
	10	11,7743	0,000600189
	11	9,61177	0,00193256
	12	9,11467	0,00253474
	13	13,4085	0,000250313
	14	12,6764	0,000370042

El p-valor para la categoría tiempo, permite afirmar que existe diferencia estadísticamente significativa en los estudiantes del grupo control y experimental, con relación a todas las preguntas relacionadas con la categoría proceso. En la figura 4, el grafico de medianas muestra la diferencia estadística entre los estudiantes del grupo control y experimental, con relación a la pregunta 13 (p13): estima el tiempo que destinará para cada etapa y el tiempo total cuando programa.

Figura 4. Diagrama de medianas, pregunta 13



Fuente: Elaboración propia

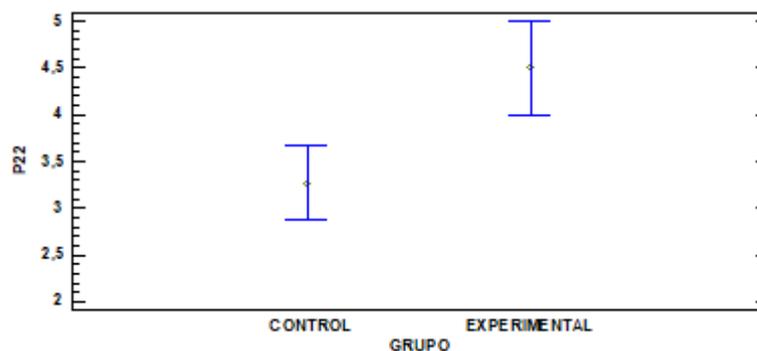
En la tabla 9, se presenta el análisis de homogeneidad para la categoría defectos.

Tabla 9. Análisis de homogeneidad categoría defectos

Categoría	Pregunta	Estadístico de prueba	p-valor
Defectos	20	11,0629	0,000880273
	21	12,9348	0,000322307
	22	12,8475	0,000337696
	23	10,71	0,00106505
	24	12,7785	0,00035038
	25	12,4771	0,000411706
	26	10,9027	0,000959748
	27	13,6782	0,000216808

El p-valor para la categoría defectos, permite afirmar que existe diferencia estadísticamente significativa en los estudiantes del grupo control y experimental, con relación a todas las preguntas relacionadas con la categoría defectos. En la figura 5, el gráfico de medianas muestra la diferencia estadística entre los estudiantes del grupo control y experimental, con relación a la pregunta 22 (p22): Tiene identificados los errores que más comete al construir software.

Figura 5. Diagrama de medianas, pregunta 22



Fuente: Elaboración propia

La prueba de hipótesis, generó para todas las preguntas de las categorías: tamaño y planeación, un p-valor  $> 0.05$ , a partir de lo cual se puede afirmar que existe diferencia estadísticamente significativa para estas categorías entre los grupos control y experimental, al final del proceso de formación.

## 6. Conclusiones

Los estudiantes inicialmente son reacios a trabajar con PSP, dado que no lo conocen y sienten que no le da un valor agregado a su labor, de allí que abordar el desarrollo de estas habilidades, deba ser un proceso cuidadosamente analizado teniendo en cuenta el currículo y los hábitos de codificación que tengan los estudiantes de programación.

Adicional a la incorporación de las buenas prácticas PSP en el curso, se tuvo un desafío con respecto a la utilización de la metodología de proyectos formativos, dado que ni el profesor titular de la materia, ni los estudiantes habían trabajado previamente de esta forma. La orientación al profesor se basó en la entrega de las guías instruccionales del curso, que se crearon para cada sesión de trabajo con los estudiantes, el material teórico, los instrumentos de recolección de datos, las actividades propuestas para los estudiantes y las rúbricas.

Los proyectos a desarrollar fueron los sugeridos por el profesor titular del curso. Estos se adaptaron a la metodología de proyectos formativos, con el fin de solicitar los productos de trabajo que finalmente fueron evaluados, sin embargo, tener diferentes proyectos durante el curso no permitió generar un proyecto evolutivo realizado por ciclos donde se incorporaran paulatinamente las prácticas sugeridas.

La experiencia de este trabajo también puede considerar un acierto la selección de los grupos control y experimental, donde sus estudiantes se encuentran en semestres intermedios de su carrera. Fueron evidentes las competencias ya desarrolladas en el campo de la programación por parte de los estudiantes de grupo experimental, lo que les permitió dedicar parte de su esfuerzo en aprender de PSP, y no sólo enfocarse en adquirir habilidades técnicas. El grupo control, con sólo un semestre adicional en su formación, era bastante similar al grupo sobre el cual se realizó la intervención.

En algunas ocasiones los estudiantes expresaron su inconformidad con el registro de los datos en los formatos sugeridos, dado que debían cumplir tanto con los requerimientos funcionales como con el desarrollo de las buenas prácticas de desarrollo de software, sin embargo, concluían que realizar los registros les permitía conocer su proceso y buscar la mejora del mismo.

El profesor titular de la materia manifestó que los estudiantes cumplieron con la entrega de los requerimientos funcionales de los proyectos asignados. No obstante, algunos puntos funcionales solicitados no se entregaban, pero no hay evidencias que permitan determinar que fuesen consecuencia del desarrollo de las buenas prácticas sugeridas para cada proyecto.

Se trabajaron prácticas y conocimientos agrupados en 5 categorías, y en todas se evidenciaron mejoras tanto teóricas como prácticas, en algunas en mayor medida que en otras. También se tuvieron buenos resultados relacionados con el compromiso con la calidad de sus proyectos, que fueron evidentes en las prácticas que realizaron.

Para la mayoría de los estudiantes se trató de su primera experiencia con PSP, porque si bien manifestaron en la encuesta de buenas prácticas de programación que se realizó al inicio de la investigación, que llevaban a cabo algunas prácticas, lo sostenían porque de manera intuitiva las realizaban, mas no tenían un proceso definido y disciplinado para construir software.

Posterior a la intervención, y después de ejecutar nuevamente la encuesta de buenas prácticas, pero en esta ocasión solamente con los grupos experimental y control, se realizó el análisis de los resultados comparando la incorporación de las prácticas PSP en el proceso de desarrollo de software de los estudiantes, encontrando que el grupo experimental tiende a utilizar más prácticas que el grupo control. También se evidenciaron diferencias entre las prácticas realizadas por los estudiantes del grupo experimental antes y después de la intervención, aumentando la ejecución de las prácticas al momento de construir software en sus labores académicas.

## 7. Referencias bibliográficas

- [1] W. Humphrey, *A discipline for software engineering*. Addison-Wesley, 1995.
- [2] D. E. Soto and A. X. Reyes, “Introduciendo PSP (Proceso Personal de Software) en el aula,” *Rev. Colomb. Tecnol. Av.*, pp. 1–5, 2010.
- [3] R. D. Rincón, “Análisis y capitalización de las experiencias y lecciones aprendidas de la implementación de PSP ( Personal Software Process ) y TSP ( Team Software Process ) desde el sector académico a las empresas de software mexicanas,” Universidad EAFIT, 2010.
- [4] J. F. N. Manrique and R. Anaya, “Estudio empírico de aplicación de PSP para el desarrollo transversal de competencias de gestión, en estudiantes de un programa de tecnología de sistemas (Trabajo fin de máster),” Universidad EAFIT, 2012.
- [5] S. A. Cardona, “Diseño de una estrategia de aprendizaje para implementar prácticas de PSP y TSP en cursos básicos de programación. Caso programa de Ingeniería de Sistemas y Computación de la Universidad del Quindío (Trabajo fin de máster),” Universidad EAFIT, 2011.
- [6] M. Towhidnejad and T. Hilburn, “Integrating the Personal Software Process (PSP) across the Undergraduate Curriculum,” in *Frontiers in Education Conference, 1997. 27th Annual Conference. Teaching and Learning in an Era of Change*, 1997, pp. 162–168.
- [7] P. Abrahamsson and K. Kautz, “Personal Software Process: Classroom Experiences from Finland,” *Lect. notes Comput. Sci.*, vol. 2349, pp. 175–185, 2002.
- [8] L. Prechelt and B. Unger, “An Experiment Measuring the Effects of Personal Software Process (PSP) Training,” in *IEEE Transactions on Software Engineering*, 2001, pp. 465–472.
- [9] W. L. Honig, “Teaching Successful ‘Real-World’ Software Engineering to the ‘Net’ Generation: Process and Quality Win!,” in *21st Conference on Software Engineering Education and Training*, 2008, pp. 25–32.
- [10] K. Venkatasubramanian, S. B. T. Roy, and M. V Dasari, “Teaching and Using PSP in a Software Engineering course: An Experience Report,” in *Software Engineering Education and Training Annual Conference*, 2001.

- [11] K. Venkatasubramanian, S. B. T. Roy, and M. V Dasari, “Teaching and Using PSP in a Software Engineering course : An Experience Report,” in *Software Engineering Education and Training Annual Conference*, 2001.
- [12] S. Tobón, *Formación integral y competencias*, Tercera Ed. Bogotá: ECOE Ediciones, 2013.
- [13] Ministerio de Industria, Turismo y Comercio, “Decreto 1377 - Régimen general para la protección de datos personales.” Bogotá, D.C., p. 11, 2013.