Artículo **Revista de Tecnología y Educación**
Diciembre, 2019 Vol.3 No.10 1-6

Técnicas para la Enseñanza de Desarrollo de Software en Alumnos de Educación Superior

Techniques for Software Development Teaching for Undergraduate Level Students

MARTÍNEZ-MIRELES, Josue†, GARCÍA-MÁRQUEZ, Marco, ESPEJEL-FLORES, Porfirio y RODRÍGUEZ-FLORES, Jazmín*

Universidad Politécnica de Pachuca, Dirección de Ingeniería en Software, Coordinación de la Maestría en Tecnologías de la Información y Comunicaciones

ID 1er Autor: *Josue, Martínez-Mireles* / **ORC ID**: 0000-0002-3313-0325, **CVU CONACYT ID**: 172614

ID 1er Coautor: *Marco, García-Márquez* / **ORC ID**: 0000-0002-3167-5209, **CVU CONACYT ID**: 180127

ID 2do Coautor: *Porfirio, Espejel-Flores* / **ORC ID**: 0000-0002-0644-4839, **CVU CONACYT ID**: 163012

ID 3er Coautor: *Jazmín, Rodríguez-Flores* / **ORC ID**: 0000-0002-7493-2762, **CVU CONACYT ID**: 321430

DOI: 10.35429/JTAE.2019.10.3.1.6 Recibido: 24 de Septiembre 2019; Aceptado: 10 de Diciembre, 2019

Resumen

La educación se ha vuelto más compleja, derivado a los entornos donde se desarrollan, los estudiantes de nivel licenciatura se caracterizan por tener diferentes tipos de aprendizaje, generando grupos heterogeneos. Es por ello que al momento de enseñarles desarrollo de software se vuelve complejo porque no todos tienen las mismas habilidades. El estudio se desarrolló en la Universidad Politécnica de Pachuca, analizando los resultados en dos grupos de la asignatura de Arquitectura Orientada a Servicios. Consistió en la organización y seguimiento de proyectos de desarrollo de software. Para ello se implementaron diferentes técnicas como la metodología de desarrollo de software, los formatos utilizados para el seguimiento del proyecto. Se consideró el tipo de Aprendizaje, con base en la aplicación de instrumentos de Programación Neurolinguistica (PNL), para identificar los estilos que tienen los alumnos de este ciclo de formación así como la evaluación diagnóstica realizada. Se pudo establecer analizar la cantidad de horas invertidas de cada alumno con respecto a la relación de tiempo total del equipo de desarrollo.

Enseñanza, Desarrollo de software, Educación superior

Abstract

Undergraduate level education has become more complex, due to the environments where the undergraduate students have grown, they develop several kinds of learning styles, generating heterogeneous groups. Software development teaching is very difficult because of that, specifically, it becomes complex because not everyone has the same skills. The study was developed at the Polytechnic University of Pachuca, analyzing the results in two groups of students of the Service Oriented Architecture course. It consisted in the organization and monitoring of software development projects, different techniques were implemented, such as: the software development methodology, project administration formats, learning style diagnosys, based on the application of Neurolinguistic Programming Instruments (NLP) to students of third year. The time invested by each student with respect to the total time ratio of the development team, is a key factor to be considered during project assignment and assessment. It's shown the results of the application of differents teaching techniques during the learning process.

Teaching, Software development, Undergraduate education

Citación: MARTÍNEZ-MIRELES, Josue, GARCÍA-MÁRQUEZ, Marco, ESPEJEL-FLORES, Porfirio y RODRÍGUEZ-FLORES, Jazmín. Técnicas para la Enseñanza de Desarrollo de Software en Alumnos de Educación Superior. Revista de Tecnología y Educación. 2019. 3-10: 1-6

* Correspondencia del Auto (Correo electrónico: jrodriguez@upp.edu.mx)
† Investigador contribuyendo como primer autor.

Introducción

Existen diferentes teórias conductuales y de aprendizaje sin que alguna pueda ser identificada como la mejor o la más efectiva, cada persona tiene su propio desarrollo cognitivo, sin embargo todas ofrecen herramientas que permiten adecuar la manera en que los docentes imparten sus asignaturas para que los estudiantes logren aprendizajes significativos. En este artículo se presentan los resultados alcanzados al aplicar diferentes técnicas en la enseñanza de la asignatura de Arquitectura Orientada a Servicios, de la Ingeniería en Software de la Universidad Politécnica de Pachuca la cual se oferta en periodos cuatrimestrales.

El estudio fue realizado con dos grupos de noveno cuatrimestre que desarrollaron proyectos de software funcional. Para llegar a la implementación exitosa de estos proyectos, se realizó una evaluación diagnóstica de los alumnos donde se identificó el conjunto de habilidades y estilos de aprendiza de los estudiantes, buscando probar la hipótesis: Es posible establecer técnicas para mejorar los aprendizajes de una asignatura de desarrollo de software en educación superior.

El validar esta hipótesis es útil porque no es fácil delimitar proyectos de software que puedan realizarse en una asignatura para periodos cuatrimestrales, con la finalidad de que terminen el proyecto en el tiempo establecido. Se usaron técnicas como la Programación Neurolinguistica (PNL) para el diagnóstico inicial de los alumnos, así mismo se aplicaron metodologías ágiles para el desarrollo de los proyectos escolares, acompañadas de formatos estandarizados para el seguimiento y admistración de proyectos.

En el marco teórico se describen los conceptos tratados en este artículo, las características del diagnóstico, las metodologías de desarrollo, los formatos usados para la administración y gestión de proyectos. En el desarrollo se muestran los resultados obtenidos sobre los tiempos de desarrollo individual y por equipo.

Marco Teórico

En esta sección se desarrollan algunos conceptos necesarios para entender mejor el artículo. Se realizó un diagnóstico del estilo de aprendizaje por ciclo de formación.

Estilos de aprendizaje

En Gamboa (2015) se realizó una caracterización de los estilos de aprendizaje y canales de percepción de estudiantes universiarios. Se utilizaron los canales de percepción (CP) Visual, Auditivo y Kinestésico. Y se realizó una caracterización de cada tipo. Siendo los Visuales quienes aprenden con imágenes, cuidan su apariencia; los auditivos caracterizados por aprender a través de escuchar con predominancia a la expresión oral, los kinestésicos o cinestésicos quienes tienen un aprendizaje vivencial.

Ciclo de formación

Para el fin de este artículo comprende la división de un programa de estudio. Su duración es de 3 periodos cuatrimestrales. Un programa educativo tiene 3 ciclos de formación que inician al 1, 4, 7 cuatrimetres.

Team Software Process (TSP)

En Humphrey, Watts. (2000) se describe cómo desarrollar software con calidad mediante una metodología de desarrollo de software basada en procesos, esta metodología abarca el contexto del desarrollo de software en equipo a través de una serie de formatos y controles de calidad mediante el uso de la estadística siempre enfocada a la mejora continua.

Personal Software Process (PSP)

En Humphrey, Watts. (2000)b. Se definen los lineamientos para el desarrollo de software individual, organizado en varios niveles que para alcanzar un proceso de desarrollo de software individual. Con formatos para el control de tiempo para llevar un proceso de aseguramiento de la calidad continuo.

Estándar IEEE 830-1998

IEEE Recommended Practice for Software Requirements Specifications. IEEE830(1998). Es un estándar para la especificación de requerimientos de software donde se plantean los requisitos funcionales, no funcionales, se clasifican en prioridades (alta, media, baja), se describe la fuente de los requsitos por mencionar algunas características.

ISSN 2523-0360

Estándar IEEE 829

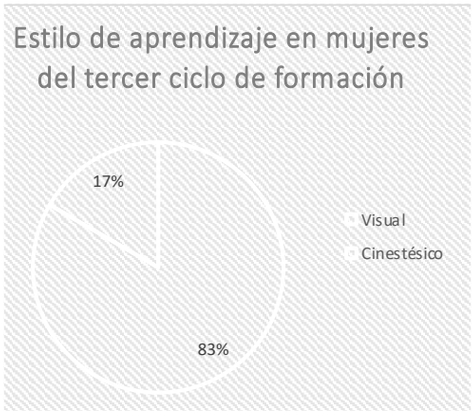
El estándar IEEE829(1998) es un estándar para pruebas de software donde se definen los casos de prueba, escenarios a través de un proceso de pruebas.

Desarrollo de la investigación

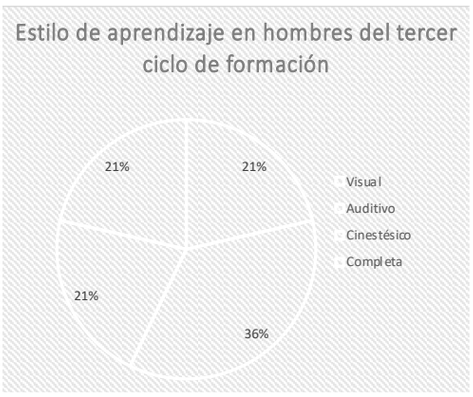
Técnica de Evaluación Diagnóstica

La evaluación diagnóstica realizada consistió en una revisión de los conocimientos previos necesarios para el desarrollo de las competencias de la asignatura (CP), los conocimientos de los contenidos temáticos (CT), las habilidades técnicas (HT) que ya poseían, las espectativas de la asignatura (EA). En los CP se identificó que los alumnos tenían deficiencias en programación orientada a objetos. En CT apenas tenían conocimiento de los principales conceptos de la asignatura.

En cuanto a HT el grupo 1 tenía más habilidades en desarrollo de software que el grupo 2. Las habilidades predominantes eran en desarrollo web. En cuanto a EA la mayoría deseaba aprender servicios web. También se identificó el Estilo de Aprendizaje (SA) obtenido por ciclo de formación y sexo para ello se usó el cuestionario propuesto por De la Parra (2004). Los resultados del SA obtenidos en las mujeres se describen en la gráfica 1 y en la gráfica 2 los hombres. Se puede observar que las mujeres del tercer ciclo de formación tienen como SA Visual, mientras que los hombres tienen un SA Auditivo. En los hombres se identifica mayor variedad en los estilos. Un 21% de los hombres tienen la predominancia de 2 estilos que algunos autores denominan completo.



Gráfica 1 Porcentaje de SA de las alumnas mujeres en el tercer ciclo de formación. Cuatrimestre del PE de Ingeniería en Software
Fuente: Elaboración Propia



Gráfica 2 Porcentaje de SA de los alumnos tercer ciclo de formación. Cuatrimestre del PE de Ingeniería en Software
Fuente: Elaboración Propia.

La evaluación diagnóstica permite establecer un punto de partida en el aprendizaje de los alumnos. Con la evaluación de CP, CT, HT, EA se puede diagnosticar a los alumnos para implementar actividades de mejora. Estas actividades se realizan al inicio de la asignatura. Las actividades de mejora realizada consistieron en un curso remedial de programación orientada a objetos de 4 sesiones.

Técnicas para el manejo y aprendizaje de la información teórica

La teoría es dificil de asimilar por los alumnos. La primera unidad de aprendizaje cuyo contenido en su mayoría es teórico, es complicada. Para lograr un aprendizaje significativo, se les asignó 8 materiales en formato pdf con los conceptos introductorios y se realizó una guía de preguntas que respondieron los alumnos. La guía de preguntas se consideró a razón de 5 preguntas por cada tema de aproximadamente 3 hojas. Con ello se logra mayor cobertura.

Se debe hacer un reforzamiento en clase con el repaso de la guía de preguntas. La viabilidad de esta técnica se probó a partir de un examen teórico en el cual se obtuvieron los resultados del Gráfico 3 del grupo 1 con 24 alumnos y el Gráfico 4 con 19 alumnos.

Cabe destacar que el examen que se aplicó fue elaborado en GoogleForms el cual a su vez se asignó en Google Classrooom. El promedio de calificaciones del grupo 1 fue de 70 y el promedio del grupo 2 fue de 74. Ambos resultados fueron aprobatorios.

ISSN 2523-0360

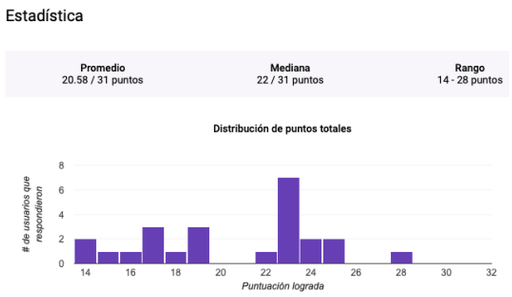


Gráfico 3 Resultados del examen teórico despues de usar la técnica de cuestionario y reforzamiento del grupo 1
Fuente: Elaboración Propia

Técnicas para el manejo y aprendizaje de la práctica en la asignatura

Para realizar la evaluación de los conocimientos prácticos se realizó un proyecto final, el cual consistió en un proyecto de software.

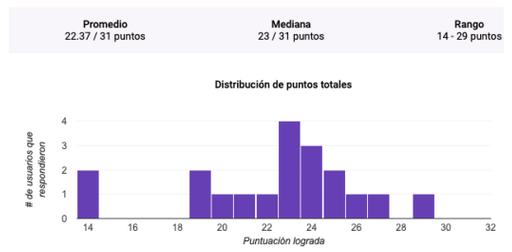


Gráfico 4 Resultados del examen teórico despues de usar la técnica de cuestionario y reforzamiento del grupo 2
Fuente. Elaboración Propia

Se especificaron las etapas de análisis, diseño, codificación y pruebas adoptando cualquier metodología de desarrollo de software que ellos eligieran.Para tener el control de los materiales generados se implementó una lista de cotejo donde se evaluó la implementación de formatos, autoevaluación y responsabilidad de los alumnos al entregar. En el desarrollo del proyecto se usaron formatos obligatorios en la documentación.

Un formato basado en el IEEE830 para la especificación de requerimientos, un formato simplificado basado en el IEEE829 para realizar las pruebas del sistema. De acuerdo a la metodología utilizada se usaron Historias de Usuario (HU) o Casos de Uso de UML (CU) para la representación de requisitos. Se usaron versiones simplificadas de los formatos de TIME RECORDING LOG (de PSP) como formatos para el registro de tiempo y el TSP MEETING REPORT FORM (de TSP) para el registro de acuerdos en las reuniones de los diferentes equipos de trabajo.

Se puede considerar que la totalidad de los proyectos fue concluido. El 28% de los proyectos se implementó con un cliente real. A continuación se describen los principales hallazgos en el experimento realizado. Con el formato de Time Recording Log fue posible identificar que cada alumno dedicó en promedio 35.9 horas al proyecto de desarrollo siendo el valor mínimo 11 y el mayor 95. Con una desviación estándar de 29.43. Como se puede apreciar en el Gráfico 5. El tiempo promedio para el desarrollo del proyecto fue de 148.48 horas por equipo. Con una desviación estándar de 93.69. como se puede apreciar en el Gráfico 6.

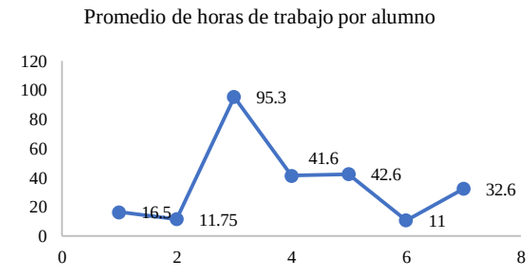


Gráfico 5 Tiempo promedio dedicado por alumno para el desarrollo de software
Fuente: Elaboración Propia

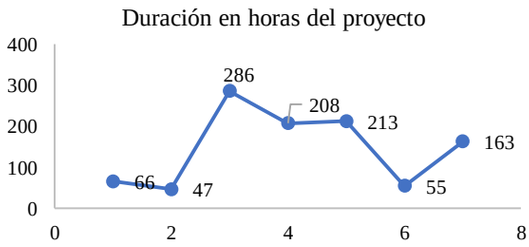


Gráfico 6 Tiempo total para el desarrollo de software por equipo
Fuente: Elaboración Propia

Descripción de proyectos desarrollados

Los proyectos desarrollados consistieron en sistemas basados en servicios web con aplicación de libre elección. Los proyectos propuestos por los alumnos fueron: 2 sistemas médicos, 2 puntos de venta, un sistema para el control de eventos, un sistema para el control de empleados y un sistema para reserva de habitaciones. El proyecto de reserva de habitaciones fue el más costoso en cuanto a tiempo se refiere con una duración total de 286 horas contra las 55 horas del sistema de control de empleados.

ISSN 2523-0360

Lenguaje de programación

Al ser un proyecto con arquitectura basada en servicios web el lenguaje predominante elegido por los alumnos fue PHP para el desarrollo del frontend, para el desarrollo de los servicios predominó el uso de Java en el backend. Algunos de los lenguajes usados fueron JavaScript, Ruby, C#.

Metodología de desarrollo (MDS)

La metodología MDS que predominó para el desarrollo de los diferentes proyectos fue Scrum, que no obliga a una documentación exhaustiva, sin embargo como una de las técnicas usadas fue el uso de variantes de los formatos PSP y TSP, se pudo llevar el control y avance del proyecto. Así mismo validar que cada uno de los alumnos participó en el desarrollo del proyecto mediante un *desarrollo horizontal*, es decir planteamiento de requisitos, diseño, desarrollo y testing . Y no se realizó un *desarrollo vertical* que consiste en que un alumno hace el documento, otro programa y otro hace pruebas. Este último tipo no logra un aprendizaje integral y significativo.

Historial de Revisiones de los Documentos de los Diversos Proyectos

Los documentos que acompañaron al proyecto final tuvieron un promedio de 2.5 revisiones, una en la etapa incial donde se definían los alcances del proyecto y otra al mes de iniciado el proyecto donde se revisó el desarrollo de los primeros módulos. Ambas revisiones fueron fundamentales pues tuvieron que ser específicas y detalladas para lograr que el proyecto se concluyera en tiempo y forma. Con el uso combinado de formatos se pudo lograr que los proyectos tuvieran un desarrollo horizontal y en consecuencia se logrará una participación de todos los alumnos involucrados.

Conclusiones

La evaluación diagnóstica es fundamental para trabajar con los grupos de educación superior. La evaluación de las competencias de la asignatura (CP), los conocimientos de los contenidos temáticos (CT), las habilidades técnicas (HT) que ya poseían, las espectativas de la asignatura (EA). Permitirán establecer actividades de mejora que permitan alcanzar las competencias en la asignatura. Las HT son fundamentales y porque permiten a los alumnos hacer tangible el proyecto en un sistema de información.

La teoría es más sencilla de asimilar si se realiza un ejercicio de validación de conocimiento de la información como un cuestionario dirigido. Como lo muestran los resultados obtenidos en ambos grupos. En este experimento se logró más de un 70% de aprobación de los conceptos teóricos.

La evaluación de la práctica en una asignatura de desarrollo de software debe establecer diferentes instrumentos para la gestión de proyectos. El registro de tiempos personal y la minuta de acuerdos son importantes para lograr un *desarrollo horizontal*.

El tiempo promedio de desarrollo de software que se puede considerar para un periodo cuatrimestral es de 148 horas. Por lo cual realizar un proyecto que exceda esa cantidad de horas correrá el riesgo de no completarse. En promedio los alumnos deberán dedicar un promedio de 35.9 horas totales al desarrollo de proyecto que dividias en 15 semanas, los alumnos deberán usar 2.39 horas a la semana para lograr un proyecto exitoso. Estos valores permiten establecer una medida en la complejidad de los proyectos de software que se pueden asignar a los alumnos en peridos cuatrimestrales. Pero además el uso de formatos como el IEEE830, Time Recording Log y el IEEE 829, permitirán una mejora en la ejecución y calidad en los procesos.

Agradecimientos

A la Universidad Politécnica de Pachuca por el apoyo recibido para el desarrollo de este artículo.

Referencias

De la Parra, E. (2004). Herencia de vida para tus hijos crecimiento integral con técnicas de PNL. México, DF. Ed. Grijalbo.

Gamboa Mora, María Cristina, & Briceño Martínez, John Jairo, & Camacho González, Johanna Patricia (2015). Caracterización de estilos de aprendizaje y canales de percepción de estudiantes universitarios. [fecha de Consulta 6 de Octubre de 2019]. ISSN: 1012-1587. Disponible en: https://www.redalyc.org/articulo.oa?id=310/31045567026

Humphrey, Watts. (2000). The Team Software Process (TSP) (CMU/SEI-2000-TR-023). Retrieved August 30, 2019, from the Software Engineering Institute, Carnegie Mellon University website: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5287

Humphrey, Watts. (2000)b. The Personal Software Process (PSP) (CMU/SEI-2000-TR-022). Retrieved August 30, 2019, from the Software Engineering Institute, Carnegie Mellon University website: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=5283

IEEE 830-1998 - IEEE Recommended Practice for Software Requirements Specifications. [fecha de Consulta 30 de Agosto de 2019]. website: https://standards.ieee.org/standard/830-1998.html

IEEE 829-1998 - IEEE Standard for Software Test Documentation [fecha de Consulta 30 de Agosto de 2019]. website: https://standards.ieee.org/standard/829-1998.html

ISSN 2523-0360