Analizando la Efectividad del Uso de un EVCI para Asistir a Estudiantes Avanzados en la Identificación de Faltas en el Código: Un Experimento Controlado

Juan P. Ucán-Pech, Raúl A. Aguilar-Vera, Antonio A. Aguileta-Güemez, Julio C. Díaz-Mendoza
Facultad de Matemáticas
Universidad Autónoma de Yucatán
Mérida, Yucatán, México
{juan.ucan, avera, aaguilet, dmendoza}@correo.uady.mx

***Resumen* — El objetivo de esta investigación se centra en el estudio de la detección de faltas con y sin apoyo de un Entorno Virtual Colaborativo Inteligente (EVCI), a través de una replicación independiente de un experimento controlado. A diferencia de otros estudios realizados con el apoyo de un EVCI, este trabajo se desarrolló empleando como sujetos experimentales a estudiantes avanzados de pregrado. En este trabajo se explora la efectividad en la detección de faltas en programas instrumentados en Java con y sin apoyo de un EVCI. Con respecto a las faltas observadas por los sujetos, se obtuvo una efectividad equivalente para quienes emplearon el EVCI (53.70%) como para quienes trabajaron de manera tradicional (50.00%). Se observa que en esta segunda réplica, los sujetos lograron identificar un número mayor de faltas con respecto a la primera réplica de este experimento.**

***Palabras clave* — Aprendizaje Colaborativo Apoyado por Computadora; Trabajo Cooperativo Apoyado por Computadora; EVCI; Aprendizaje de la programación; Experimentación en Ingeniería de Software.**

I. Introducción

La detección de faltas ha sido un área de creciente interés en investigadores, particularmente en el área de ingeniería de software.

Esta investigación inicio con una replicación de un experimento [1], en este artículo se extiende ese estudio con un segundo experimento.

En esta segunda réplica se analiza la efectividad en la detección de faltas en programas instrumentados en Java con y sin apoyo de un EVCI, pero ahora empleado sujetos avanzados de pregrado. Este documento se estructura de la siguiente manera: en la sección II se describen los antecedentes a esta investigación; en la sección III se muestra el proceso general de experimentación de una segunda réplica del experimento. Finalmente en la sección IV se describen las discusiones y conclusiones.

II. Antecedentes

A. Entorno Virtual Colaborativo inteligente

El significado de EVCI reside en la combinación de dos áreas de investigación como son los Entornos Virtuales Colaborativos [2] y la incorporación de técnicas de Inteligencia Artificial [3].

La aplicación de software, desarrollada bajo el dominio de EVCI que se ha estado utilizando en esta línea de investigación, permite a los usuarios visualizar trozos de código de programas implementados bajo un paradigma de programación, en este artículo se explora con el paradigma de Programación Orientado a Objetos.

El software es una aplicación Web que permite a los estudiantes encontrar faltas en una serie de programas instrumentados, los estudiantes organizados en grupos de trabajo pueden realizar dicha actividad consultado con el profesor o con el componente inteligente que consiste en un Sistema Experto (SE) para identificar los tipos de faltas comunes.

La arquitectura implementada en la aplicación Web consiste en una arquitectura de tres capas.

La capa de presentación, donde se encuentra la interfaz del sistema. Se trata de una aplicación Web que le permite a cada uno de los profesores por medio del SE, hacer el modelado de los errores comunes de la programación.

La capa de lógica de negocio está dividida en tres partes. El módulo de gestión de información del Entorno Virtual Colaborativo (EVC), la lógica KB (base de conocimiento, por sus siglas en inglés Knowledge Base: KB) y el motor de inferencia, siendo éstos dos últimos submódulos del SE.

En la capa de datos se encuentra la base de conocimiento que se ve afectada por los dos submódulos correspondientes al SE. El módulo de Lógica KB afecta a la Base de Conocimiento por medio de los cambios solicitados en los errores y diagnósticos de errores asociados a cada tipo de falta al hacer el modelado.

B. Estudio replicado

Esta investigación fue motivada por experimentos que comparan el estudio de detección de faltas con y sin apoyo de un Entorno Virtual Comparativo Inteligente [1].

En este experimento, se emplearon como sujetos experimentales a 46 estudiantes que estaban finalizando el primer semestre de su primer año de licenciatura, de acuerdo a la clasificación sobre experiencia en programación creada por [4], esos estudiantes se clasifican como principiantes. Los sujetos estuvieron organizados en equipos de dos y tres estudiantes, en total se organizaron en 18 equipos.

El diseño usado fue un diseño cruzado (en inglés, Cross Over) con el fin de obtener un mayor número de observaciones [5], en específico se usó un diseño cruzado 2 x 2, dos tratamientos (sujetos Con EVCI [C], y sujetos Sin EVCI [S]) en dos periodos distintos, tal como se ilustra en la Tabla 1.

Como artefactos, además del EVCI, los sujetos usaron dos programas en lenguaje C, (matriz.c y alumnos.c) [3], a cada programa se le inyectó siete faltas.

TABLA 1. DISEÑO CRUZADO 2 X 2 EMPLEADO EN EL EXPERIMENTO

	Secuencia 1	Secuencia 2
Sesión 1	C	S
Sesión 2	S	C

Una de las variables de estudio en la investigación fue la efectividad, los resultados obtenidos en la primera réplica, sugirieron una efectividad equivalente en la detección de faltas para quienes emplearon el ECVI como para quienes trabajaron de manera tradicional.

En cuanto a la tipología de la replicación empleada en esta segunda réplica, de acuerdo con [6] sería una replicación Literal-Conjunta-Interna. La replicación independiente hecha fue posible por la existencia de un kit experimental proporcionada por los investigadores originales.

III. CONTEXTO DE LA REPLICACIÓN

En esta sección, ahora se describe la información de la nueva replicación, se realizó considerando como referencia el proceso general de experimentación en ingeniería de software descrito en [7].

A. Definición

Como se expuso anteriormente en [1], esta investigación tiene como finalidad analizar a través de un experimento controlado la efectividad del aprendizaje en atención a la detección de faltas en programas de software mediante el uso de un EVCI; y para lograr este objetivo se han planteado las siguientes hipótesis:

> ***H_0***: La efectividad medida como el porcentaje de faltas observadas por los sujetos es igual tanto para los sujetos que utilizan el EVCI como para los sujetos que trabajan sin el empleo del EVCI.
>
> ***H_1***: La efectividad medida como el porcentaje de faltas observadas por los sujetos es diferente entre los sujetos que utilizan el EVCI y los sujetos que trabajan sin el empleo del EVCI.

En este contexto, la efectividad es medida con respecto al porcentaje de faltas encontradas del total de faltas inyectadas en un programa de software.

B. Diseño

En esta segunda replica se mantiene el diseño cruzado [5]. Tal como se ha mencionado, en este diseño, las unidades experimentales son los grupos de sujetos conformados por tres estudiantes que reciben dos o más tratamientos, y el orden de aplicación de los tratamientos está determinado por la estructura de este diseño. En la Tabla 1 se muestra la estructura del diseño experimental empleado.

En esta estructura, en el primer periodo, una mitad de grupos de sujetos recibe el tratamiento C mientras que la otra mitad de grupos recibe el tratamiento S; en el segundo periodo los grupos de sujetos reciben los tratamientos en orden inverso.

Como estrategia para mitigar el inconveniente del efecto remante por extenderse más allá del periodo de aplicación, se consideró no aplicar los tratamientos en periodos consecutivos, en nuestro caso se planificó un descanso de 3 días entre ambos periodos.

TABLA 2. DISEÑO CRUZADO 2 X 2 EMPLEADO EN LA SEGUNDA RÉPLICA

	Secuencia 1	Secuencia 2
Periodo 1 (Programa A)	C	S
Periodo 2 (Programa B)	S	C

C. Ejecución

En este experimento participaron 28 (22 hombres, 6 mujeres) estudiantes inscritos a la asignatura "Administración de Proyectos II" del último semestre de la Licenciatura en Ingeniería de Software correspondiente al periodo enero mayo de 2015. Se realizó el experimento en dos salas de la Facultad de Matemáticas de la Universidad Autónoma de Yucatán (UADY), una de esas dos salas fue en el centro de cómputo, allí estuvieron los estudiantes que trabajaron con el EVCI, mientras que los que trabajaron sin el EVCI la sala correspondió a un salón de clases sin equipo de cómputo.

El experimento consistió en identificar seis faltas inyectados intencionalmente en el código de dos programas de software escritos en lenguaje Java, las faltas se tomaron como referencia de la clasificación propuesta en [8]. El programa A (cuentas.java), lee comandos desde la consola y los procesa para probar una serie de funciones, las cuales permiten utilizar cuentas bancarias, las cuentas se les puede añadir dinero, obtener dinero de ellas y transferir el dinero entre ellas. El programa B (lista.java) lee comandos desde la consola y los procesa para probar una serie de funciones, las cuales permiten manejar una lista la cual puede estar en orden de mayor a menor, es decir una lista cuyos nodos tienen un número entero, la lista puede o no estar ordenada.

En relación a la asignación de las unidades experimentales a los tratamientos se organizaron equipos de dos y tres estudiantes, en total fueron 9 equipos los cuales se asignaron de forma aleatoria a alguna de las dos secuencias de tratamientos, tal como se lustra en la Tabla 2.

TABLA 3. ORGANIZACIÓN DE EQUIPOS PARA LA SEGUNDA REPLICA

	Equipos con el tratamiento [C]	Equipos con el tratamiento [S]
Periodo 1 (Programa A)	3,6,8,9	1,2,4,5,7
Periodo 2 (Programa B)	1,2,4,5,7	3,6,8,9

En cada periodo, ambos grupos del experimento trabajaron con el mismo programa, sin embargo, en el primer periodo cuatro equipos fueron provistos del empleo del EVCI (Tratamiento C) para la identificación de las faltas; mientras que a los otros cinco equipos se les proporcionó el programa de manera impresa, y por lo tanto, no utilizaron el EVCI (Tratamiento S); en el segundo periodo los grupos de sujetos recibieron los tratamientos en orden inverso.

Durante el experimento a todos los equipos se les proporcionó un documento impreso con las especificaciones del programa bajo análisis, y una plantilla impresa para el registro de la información referente a las faltas detectadas en el programa.

Es importante mencionar que previo a la aplicación del experimento, los estudiantes fueron sometidos a una sesión de dos horas durante las cuales se les explicó el funcionamiento y uso del EVCI, y los tipos de faltas comunes de programación. Posteriormente, para finalizar el proceso de capacitación y en una nueva sesión, los alumnos realizaron un ejercicio con el fin de practicar el uso de la herramienta para la detección de faltas en un programa de software.

D. Análisis

En esta sección se presenta un análisis exploratorio de los datos obtenidos en el experimento, y el análisis del modelo estadístico asociado al diseño cruzado utilizado para el estudio.

Análisis exploratorio

En la Tabla 4 que se muestra a continuación se listan las observaciones acerca de la efectividad promedio en la búsqueda de faltas.

TABLA 4. EFECTIVIDAD PROMEDIO DE LOS TRATAMIENTOS

Tratamiento	Efectividad
ECVI (c)	53.70%
Sin entorno (s)	50.00%

Listadas las estadísticas resultantes del experimento, se procede a realizar un sencillo análisis exploratorio de datos empezando con una inspección gráfica del comportamiento de los mismos. En la Figura 1 se muestra un diagrama de cajas para la efectividad en la detección de errores por cada tratamiento, donde el eje vertical representa la respuesta (efectividad) y el eje horizontal representa los tratamientos: con empleo del EVCI (C) y sin empleo del EVCI (S).

Continuando con el análisis de la Figura 1, se observa que la mayoría de las mediciones tienen el mismo valor en ambos tratamientos, con excepción de dos faltas. Con respecto a la efectividad, la mayoría de los participantes detectó el mismo número de faltas en ambos tratamientos (uso del EVCI [C], sin uso del EVCI [S]).

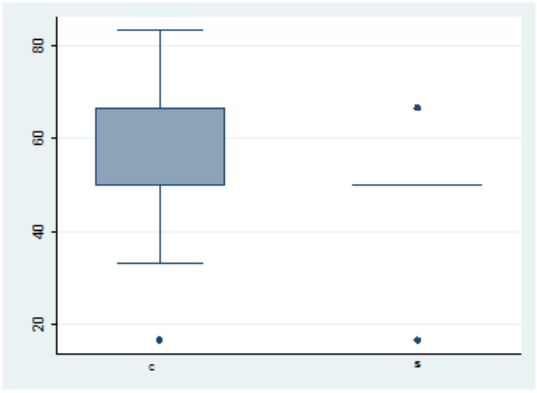

Figura 1.Diagramas de caja de los tratamientos con respecto a la efectividad

En la Tabla 5 se muestran los promedios de efectividad con respecto a los programas empleados en las dos sesiones del experimento, en promedio los sujetos lograron observar un mayor número de faltas en el programa A.

TABLA 5. EFECTIVIDAD PROMEDIO EN LA DETECCIÓN DE FALTAS CON RESPECTO AL PROGRAMA

Periodo	Efectividad
Periodo 1 (Programa A)	61.10%
Periodo 2 (Programa B)	42.59%

En la Figura 2 se muestra el diagrama de cajas de ambos programas (periodos) con respecto a la efectividad. En el caso del primer programa se observan algunos valores atípicos así como cierto grado de asimetría.

En la Tabla 6 se presentan los promedios de efectividad con respecto a las dos secuencias empleadas en este diseño experimental. En la Tabla 6 se observa que los promedios en ambas secuencias parecen no tener diferencias sustanciales por lo que pudiera intuirse la ausencia de efectos remanentes en los tratamientos.

En la Figura 3 se muestra el diagrama de cajas de ambas secuencias (1:c-s, 2:s-c) con respecto a la efectividad. En este diagrama se observa que la efectividad es similar en ambas secuencias por lo que puede intuirse la falta de posibles efectos remanentes.

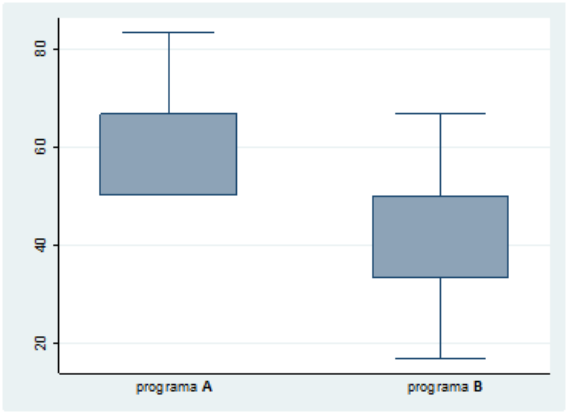

Figura 2.Diagramas de caja de los programas con respecto a la efectividad

TABLA 6. EFECTIVIDAD PROMEDIO CON RESPECTO A LAS SECUENCIAS DE APLICACIÓN DE LOS TRATAMIENTOS

Secuencia	Efectividad
1 C-S	54.16%
2 S-C	50.00%

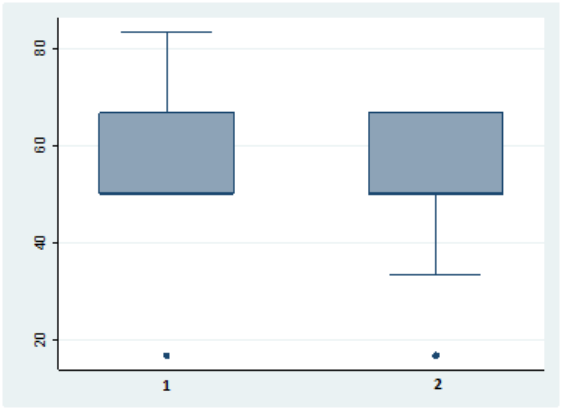

Figura 3.Diagramas de caja de las secuencias con respecto a la efectividad

Análisis del modelo estadístico

Finalizado el análisis exploratorio de los datos, se procede a probar las hipótesis del estudio. Los resultados del análisis del modelo se describen en la ecuación (1).

$$y_{ijk} = \mu + \alpha_i + b_{ij} + \gamma_k + \tau_d + \lambda_c + \varepsilon_{ijk} \quad (1)$$

Donde μ es el promedio general, αi es el efecto de la secuencia, b_{ij} es el efecto aleatorio para cada sujeto con promedio 0 y varianza σb2, γ_k es el efecto del periodo, τ_d es el efecto directo del tratamiento, λc es el efecto remanente, y ε_{ijk} es el error aleatorio independiente con promedio 0 y varianza σ2. En la Tabla 7 se muestran los resultados del análisis de varianza con respecto a la efectividad medida como el porcentaje de faltas observadas por los grupos de sujetos. Dadas las componentes analizadas en la ANOVA, se observa una diferencia significativa a un nivel α del 0.05 en el periodo (programa). Con respecto al tratamiento y al efecto remanente no se observan diferencias significativas.

TABLA 7. RESULTADOS DEL ANÁLISIS DE VARIANZA

Componente	SC Parcial	gl	SM	F	Prob > F
Tratamiento	222.20	1	222.20	0.98	0.3384
Periodo (programa)	1209.89	1	1209.89	5.35	0.0365
Efecto remanente	77.15	1	77.15	0.34	0.5685
Residuos	3166.57	14	226.18		

La medida de separación entre el tratamiento y el efecto remanente arroja un valor cercano al 30% (29.2893%). Esta medida indica el grado de ortogonalidad entre ambos efectos. Un diseño cruzado 2 x 2 como el aquí usado está expuesto a un porcentaje bajo en la medida de separación. Se recomienda emplear este diseño cuando se intuya la ausencia de efectos remanentes. Dado que no se ha observado un efecto remanente significativo los resultados del ANOVA pueden considerarse confiables.

IV. DISCUSIONES Y CONCLUSIONES

En la Tabla 8 se muestran los resultados previos que han analizado la efectividad en la detección de faltas utilizando un Entorno Virtual Colaborativo Inteligente. La efectividad es medida con respecto al porcentaje de faltas encontradas del total de faltas inyectadas en un programa de software.

TABLA 8. EFECTIVIDAD OBSERVADA EN TRABAJOS PREVIOS CON RESPECTO A LOS TRATAMIENTOS

Experimento	Efectividad con el tratamiento [C]	Efectividad con el tratamiento [S]
Ucán, Gómez y Vera (primera réplica)	43.45%	45.33%
Réplica aquí reportada	53.70%	50.00%

En la Tabla 8, se observa que los resultados aquí reportados muestran un porcentaje superior en la efectividad con respecto a los tratamientos. A diferencia de la réplica anterior, en este experimento se emplearon estudiantes de pregrado cursando su último año de carrera, mientras que en la primera réplica se emplearon estudiantes de primer año de carrera.

Con respecto a las faltas observables, se obtuvieron porcentajes mayores utilizando el EVCI que sin el EVCI en esta segunda réplica en comparación con la réplica anterior.
De manera general, se observó que los sujetos empleados en esta segunda réplica bajo el mismo contexto del experimento, lograron identificar un número mayor de faltas con respecto a los tratamientos, programas y secuencias que los observados en la primera réplica.

REFERENCIAS

[1] J. P. Ucán, O. S. Gómez, A. A. Castillo, and R. A. Aguilar. *Detección de defectos con y sin apoyo de un entorno virtual colaborativo inteligente en cursos introductorios de programación.* In O.S. Gómez, G. Arcos, L. Aguirre, E. Villa, and R. H. Rosero, editors, Ingeniería de Software e Ingeniería del Conocimiento. Jornadas Iberoamericanas. 11th (JIISIC'2015). Curran Associates, Inc., Riobamba, 2015, p. 65–78.

[2] D. Snowdon, E. F. Churchill, and A. J. Munro. Collaborative virtual environments: Digital spaces and places for CSCW: An introduction. In *Collaborative virtual environments*, Springer London, 2001, pp. 3-17.

[3] J. P. Ucán-Pech, "Aprendizaje de la Programación Asistido con Entornos Virtuales Colaborativos Inteligentes". Tesis Doctoral. Dirección de Posgrado e Investigación de la Universidad del Sur, Campus Mérida, México, 2015.

[4] H. L. Dreyfus and S. Dreyfus, "Mind over Machine. The Power of Human Intuition and Expertise in the Era of the Computer", New York: Basil Blackwell, 1986.

[5] R. O. Kuehl. *Design of experiments: statistical principles of research design and analysis*. Duxbury/Thomson Learning, 2000.

[6] O. S. Gómez, "Tipología de Replicaciones para la Síntesis de Experimentos en Ingeniería del Software". Tesis Doctoral. Universidad Politécnica de Madrid, 2012.

[7] O. Gómez, J. Ucán and G. Gómez, "Aplicación del proceso de experimentación a la ingeniería de software," Abstraction & Application, vol. 8, pp. 26–37, 2013.

[8] V. R. Basili, and R. W. Selby, "Comparing the effectiveness of software testing strategies", *IEEE Transactions on Software Engineering* 12, pp. 1278-1296, 1987.

Juan Pablo Ucán Pech es Doctor en Sistemas Computacionales por la Unidad de Posgrado e Investigación de la Universidad del Sur, México. Maestro en Sistemas Computacionales con especialidad en Ingeniería de Software por el Instituto Tecnológico de Mérida, México. Licenciado en Ciencias de la Computación por la Facultad de Matemáticas de la Universidad Autónoma de Yucatán (UADY), México. Actualmente es Profesor Titular en la Facultad de Matemáticas e integrante del Grupo Académico de Tecnologías para la Formación en Ingeniería de Software de la UADY. Su trabajo de investigación se centra en temas relacionados con la Ingeniería de Software, Ingeniería Web e Informática Educativa.

Raúl Antonio Aguilar Vera es Licenciado en Ciencias de la Computación por la Universidad Autónoma de Yucatán (UADY), Doctor en Informática por la Universidad Politécnica de Madrid, España. Actualmente es profesor de tiempo completo en la Facultad de Matemáticas e integrante del Grupo Académico de Tecnologías para la Formación en Ingeniería de Software de la UADY. Su trabajo de investigación incluye las siguientes áreas: Ingeniería de Software e Informática Educativa.

Antonio Armando Aguileta es Licenciado en Ciencias de la Computación por la Universidad Autónoma de Yucatán (UADY), Maestro en Ciencias Computacionales por el Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), campus Monterrey. Actualmente es Profesor Asociado en la Facultad de Matemáticas e integrante del Grupo Académico de Tecnologías para la Formación en Ingeniería de Software de la UADY. La línea de investigación de su interés es en torno a la calidad en la Ingeniería de Software.

Julio César Díaz Mendoza es Ingeniero Industrial en Producción por el Instituto Tecnológico de Mérida (ITM). Maestro en Tecnologías de Información, por la Universidad Interamericana para el Desarrollo (UNID). Especialista en Docencia de la Universidad Autónoma de Yucatán (UADY). Actualmente imparte asignaturas en las Licenciaturas en Ingeniería de Software, y en Ciencias de la Computación de la Facultad de Matemáticas, relacionadas con las áreas de Ingeniería de Software. Es integrante del Grupo Académico de Tecnologías para la Formación en Ingeniería de Software de la UADY, su interés se enfoca en Procesos de Software e Ingeniería de Software Educativa.