

Alianza Europea de Innovación para la Educación del Tesing Proyecto ENACTEST

Niels Doorn¹, Fernando Pastor Ricós², Beatriz Marín², Tanja Vos^{1,2}

¹Open Universiteit (OU), The Netherlands

²Universitat Politècnica de València (UPV), Spain

niels.doorn@ou.nl, fpastor@pros.upv.es, bmarin@dsic.upv.es, tvos@dsic.upv.es

Abstract. La importancia de las pruebas de software es indiscutible, pero su falta de implementación adecuada puede provocar aplicaciones con errores y problemas. Esta situación se debe a un desajuste entre las habilidades demandadas por la industria, las necesidades de aprendizaje de los estudiantes y los métodos actuales que se utilizan para la enseñanza de las pruebas en instituciones educativas. El proyecto ENACTEST busca cubrir este desajuste mediante el desarrollo de cápsulas de enseñanza para las pruebas del software (*testing*) adaptadas a las demandas de la industria y a los procesos cognitivos de los estudiantes. El consorcio incluye universidades, centros de formación profesional y pequeñas empresas, que representa un entorno socioeconómico amplio.

1. Introducción

Hoy en día, los sistemas de software se utilizan ampliamente en diversos campos, lo que amplifica las consecuencias de sus fallos, pudiendo causar daños significativos a la sociedad. Las pruebas de software son fundamentales para garantizar la calidad y minimizar los riesgos asociados con los fallos. Sin embargo, en muchas organizaciones falta una cultura de pruebas del software, ya que estas suelen ocupar un lugar secundario en el proceso de desarrollo debido a la presión por entregar actualizaciones rápidas y a la creciente complejidad de los sistemas modernos. Las pruebas del software en la industria enfrentan numerosos desafíos, como el conocimiento insuficiente de metodologías de pruebas, sesgos en los supuestos utilizados para diseñar casos de prueba y la desacertada selección de herramientas para optimizar el proceso, entre otros [Afzal et al. 2020] [Juhnke et al. 2021] [Garousi et al. 2020a].

Para preparar mejor a los profesionales de la industria, es fundamental intervenir la raíz del problema: la educación de las pruebas de software. Aunque se han realizado varios esfuerzos académicos para mejorar este ámbito, como se refleja en revisiones sistemáticas de la literatura [Scatalon et al. 2019] y [Garousi et al. 2020b], numerosos problemas aún persisten. Entre ellos, la desconexión entre la teoría y la práctica reduce el interés de los estudiantes; los ejemplos en clase están alejados de los proyectos reales; los estudiantes carecen de confianza en sus habilidades para probar el software, y la falta de experiencia práctica limita la capacidad de los profesionales novatos para llevar a cabo el proceso completo de pruebas en entornos industriales.

El proyecto ENACTEST enfrenta estos desafíos mediante el desarrollo de materiales didácticos, llamados *cápsulas*, cuya principal característica es que son breves y fáciles de integrar en los cursos existentes sin imponer una carga de trabajo adicional a

los profesores. Las cápsulas presentan técnicas de enseñanza novedosas, adaptadas a las necesidades de aprendizaje de los estudiantes y alineadas con las demandas de la industria para mejorar sus habilidades en el proceso de pruebas del software. Las cápsulas corresponden a materiales didácticos de diversa naturaleza (por ej. ejercicios, casos de estudio, juegos, herramientas, etc.), que han sido diseñados para fomentar el aprendizaje de tópicos de pruebas específicos, por ej. pruebas de mutación, pruebas de regresión, etc.

2. Objetivos del proyecto

El proyecto ENACTEST (2022-2025) tiene como objetivo crear materiales didácticos coherentes y oportunos para la enseñanza de las pruebas del software, considerando tanto las necesidades de la industria como los modelos cognitivos de los estudiantes. Para ello, en el proyecto ENACTEST se ha analizado la viabilidad de incorporar nuevas técnicas de enseñanza/aprendizaje en la educación superior, con el fin de mejorar las competencias de los estudiantes y, por ende, las habilidades de los futuros profesionales de la informática. Este propósito se puede resumir en la reducción de las brechas en la educación sobre pruebas desde tres perspectivas interconectadas en la educación de la informática: los estudiantes, la industria y el ámbito académico, como se ilustra en la Figura 1.



Figure 1. Proyecto ENACTEST (<https://enactest-project.eu>).

Para alcanzar el objetivo general de ENACTEST, se han definido los siguientes objetivos específicos:

- Identificar los modelos cognitivos que utilizan los estudiantes y los expertos cuando tienen que lidiar con pruebas, especialmente al diseñar casos de prueba.
- Categorizar las necesidades y habilidades que requiere la industria para las pruebas del software y las habilidades fundamentales que se deberían incluir en los planes de estudio.
- Diseñar y desarrollar materiales para la enseñanza de las pruebas del software (cápsulas) para incorporarlos a la educación de manera fácil. Estos materiales tendrán en cuenta los modelos cognitivos de los estudiantes y las necesidades de la industria.
- Proporcionar evidencia de mejora en los resultados de aprendizaje para los estudiantes y una mejor transferencia de conocimientos a la industria.

3. Beneficiarios

El consorcio del proyecto está compuesto por un grupo diverso de beneficiarios para garantizar que los resultados beneficien a la totalidad del panorama socioeconómico. El

consorcio incluye universidades (4), centros de formación profesional (1) y pequeñas empresas (4). Los beneficiarios del proyecto ENACTEST son: Universitat Politècnica de València (UPV), España; Katholieke Universiteit Leuven (KULeuven), Bélgica; Universidade do Porto (UP), Portugal; Università degli Studi di Napoli Federico II (UNINA), Italia; Research Institutes of Sweden (RISE), Suecia; Centro Superior de Formación Europa-Sur (CESUR), España; NEXO QA, España; INOVA+, Portugal; CTG, Bélgica.

4. Resultados

Para alcanzar los objetivos del proyecto ENACTEST, se ha realizado la especificación de requisitos, modelos e implementación de un caso para realizar las pruebas, y se ha diseñado un experimento que ha sido realizado en diferentes centros de educación superior y centros de formación profesional para descifrar el proceso cognitivo de los estudiantes cuando diseñan casos de pruebas utilizando distintas estrategias [Doorn et al. 2023]. Además, se han realizado tres grupos focales con profesionales para entender las necesidades de la industria en el proceso de pruebas y las carencias existentes en la transferencia de conocimiento en los equipos de pruebas. También, se ha llevado a cabo un mapeo de cursos donde se enseñan las pruebas del software en los países de la alianza de innovación, y se ha entrevistado a profesores para comprender los problemas y necesidades relacionados con la enseñanza de las pruebas en la práctica [Tramontana et al. 2024].

Teniendo en cuenta los resultados sobre el proceso cognitivo de los estudiantes, las necesidades de la industria y las necesidades académicas, se han diseñado las siguientes cápsulas:

1. GIPPPY: Juego en línea para la introducción temprana del testing y del pensamiento computacional [Lapeña et al. 2022] - UPV.
2. INCITE: Entrevistas semiestructuradas para enseñar testing y promover el momento A-HA en estudiantes - UPV.
3. GADGETS: Colección de juegos análogos para practicar estrategias de testing siendo conscientes de los supuestos que pueden sesgar el proceso - UPV.
4. FRAFOL: Framework para el aprendizaje de testing de mutación - UP.
5. GAMFLEW: Juego para aprender pruebas de caja blanca - UP.
6. TES Game: Juego para la detección de test smells y refactorización - UNINA.
7. Testing Robot Challenge: Herramienta para competir mediante el testing manual con testing automático en diversos desafíos - UNINA.
8. TESCAV: Herramienta para visualizar de cobertura en modeos - KU Leuven.
9. ModelDefenders: Juego para aprender testing de modelos basado en mutaciones - KU Leuven
10. Herramienta para aprender testing basado en estados - RISE.
11. BDD: Artefactos y ejercicios para aprender testing de acepción usando BDD - NexoQA.
12. Cyber: Artefactos y ejercicios para las pruebas de ciberseguridad - NexoQA.
13. TDD: Artefactos y ejercicios para el desarrollo dirigido por pruebas - CESUR.
14. AGILE: Artefactos y guías para aplicar testing en procesos ágiles - CESUR.
15. WCAG: Artefactos y ejemplos para enseñar testing de accesibilidad y WCAG - CTG.
16. TD Tech: Base de conocimiento y ejercicios de técnicas de diseño de casos de prueba - CTG.

17. DELTA: Herramienta para la detección de cambios a través de la interfaz gráfica d
elos sistemas para aprender testing de regresión y validar cambios no intencionales
que afectan la experiencia de usuario [Pastor Ricós et al. 2025].

Las cápsulas están en un proceso continuo de refinamiento y validación. Los metadatos de las cápsulas han sido definidos según los estandares IEEE LOM y AR-IANDE, donde se describe el tipo de recurso educativo (juego, entorno, ejercicio), el nivel de interacción, los prerequisitos, los objetivos de aprendizaje, su nivel de cumplimiento según la taxonomía de Bloom, el nivel de los estudiantes (grado, master, doctorado), el tema específico dentro de la disciplina del software testing, el enfoque educativo, y el acceso a la cápsula junto a los materiales (videos, poster, etc.) que la soportan. Las cápsulas junto a los materiales y recursos desarrollados serán publicados en el sitio web del proyecto ENACTEST durante el 2025. Esto permitirá que la comunidad en general pueda acceder a ellas y utilizarlas para sus fines educativos y profesionales. Finalmente, para potenciar una cultura de pruebas y reforzar la importancia de ellas a toda la sociedad, se publican noticias quincenales en el sitio del proyecto ¹.

Agradecimientos Este proyecto ha sido financiado en la convocatoria ERASMUS-EDU-2021-PI-ALL-INNO con el número 101055874, 2022-2025

References

- Afzal, A., Le Goues, C., Hilton, M., and Timperley, C. (2020). A study on challenges of testing robotic systems. In *13th ICST*, pages 96–107. IEEE.
- Doorn, N., Vos, T., and Marín, B. (2023). Towards understanding students' sensemaking of test case design. *Data & Knowledge Engineering*, page 102199.
- Garousi, V., Felderer, M., Kuhrmann, M., Herkiloğlu, K., and Eldh, S. (2020a). Exploring the industry's challenges in software testing: An empirical study. *Journal of Software: Evolution and Process*, 32(8):e2251.
- Garousi, V., Rainer, A., Lauvås Jr, P., and Arcuri, A. (2020b). Software-testing education: A systematic literature mapping. *Journal of Systems and Software*, 165:110570.
- Juhnke, K., Tichy, M., and Houdek, F. (2021). Challenges concerning test case specifications in automotive software testing: assessment of frequency and criticality. *Software Quality Journal*, 29:39–100.
- Lapeña, A., Marin, B., and Vos, T. (2022). Gippyp: Game for introductory programming practice in python. In *INTED2022 Proceedings*, pages 5598–5606. IATED.
- Pastor Ricós, F., Marín, B., Vos, T., Neeft, R., and Aho, P. (2025). Delta gui change detection using inferred models. *Computer Standards & Interfaces*, 92:103925.
- Scatalon, L., Carver, J., Garcia, R., and Barbosa, E. (2019). Software testing in introductory programming courses: A systematic mapping study. In *50th SIGCSE*, pages 421–427. ACM.
- Tramontana, P., Marin, B., Paiva, A., Mendes, A., Vos, T., Amalfitano, D., Cammaerts, F., Snoeck, M., and Fasolino, A. (2024). State of the Practice in Software Testing Teaching in Four European Countries. In *17th ICST*, pages 59–69. IEEE.

¹<https://enactest-project.eu/news/>