IE comunicaciones

Revista Iberoamericana de

Informática Educativa

ADIE

Revista Iberoamericana de
Informática Educativa

IE comunicaciones

Revista Oficial de la
Asociación para el Desarrollo de la Informática Educativa
ADIE

IE Comunicaciones es una publicación periódica editada y distribuida por la Asociación para el Desarrollo de la Informática Educativa ADIE. No se solidariza, necesariamente, con la opinión expresada por los autores de los artículos.

Edita: Asociación para el Desarrollo de la Informática Educativa, ADIE
ISSN: 1699-4574

Número 42, Julio – Diciembre 2025

Sumario

Artículos

ADIE

Fines

- Fomentar la Informática Educativa.
- Promover la formación de las personas en las Nuevas Tecnologías Educativas.
- Intercambiar trabajos, ideas y experiencias.
- Evaluar la calidad pedagógica de los productos existentes

Actividades

- Cursos, Seminarios, Mesas Redondas, Conferencias, Talleres, etc.
- Congresos periódicos de carácter nacional e internacional.
- Evaluación y Biblioteca de Software Educativo.
- Base de Datos de investigaciones, estudios sistemáticos.

Revista

IE comunicaciones

Revista Iberoamericana de Informática Educativa

Editora: Diana Pérez (diana.perez@urjc.es)

Miembros

Socios Honorarios
Socios Institucionales
Socios Individuales
Alumnos

Socios Institucionales

Universidad Nacional de Educación a Distancia
Universidad Autónoma de Madrid
Facultad de Informática (Universidad de Málaga)
Universidad Politécnica de Madrid
Dpto. De Informática y Automática, Fac. de Ciencias (Universidad Complutense de Madrid)
Universidad de Castilla-La Mancha

Web

Adie.es

Envío de contribuciones

diana.perez@urjc.es

El proyecto de finalización de carrera como una oportunidad de profundización de conocimientos. Caso de estudio: testing de software

Sonia I. Mariño, María de los A. Ferraro, Laura I. Gómez Solis, Itati Araujo, Janet García

msonia@exa.unne.edu.ar, ferraroma@exa.unne.edu.ar
Facultad de Ciencias Exactas y Naturales y Agrimensura.
Universidad Nacional del Nordeste, Argentina

Resumen: Introducción. El Proyecto Final de Carrera (PFC) constituye un desafío significativo para la culminación de los estudios universitarios en las carreras de licenciatura, en particular, en aquellas comprendidas por las Ingeniería o Ciencias de la Computación. Métodos. Se diseñó un método, que consta de fases definidas y se válida para identificar y vincular competencias específicas, sociales y profesionales de un tema tratado en una asignatura de Ingeniería de Software y profundizado en la producción realizada en el PFC. La validación se desarrolla en un contexto académico real, con la participación de estudiantes que se desempeñan laboralmente en áreas afines al objeto de estudio. Resultados. Se evidencia la validez de la metodología propuesta para identificar competencias desarrolladas durante la elaboración del PFC en relación con aspectos de testing o pruebas de software mediante el uso de herramientas específicas. Conclusiones. Se constató la integración y profundización de conocimientos técnicos y profesionales, a través de una experiencia que fortalece el enfoque por competencias y el aprendizaje centrado en el estudiante en la etapa final de la carrera. Se destaca la participación de mujeres que asumieron diversos roles —estudiantes, profesora orientadora y docentes responsables de asignaturas—, lo que contribuye a visibilizar su intervención en ámbitos técnicos y académicos de la Ingeniería de Software, promoviendo prácticas formativas más inclusivas y contextualizadas.
Palabras clave: educación superior; competencias profesionales; educación centrada en los estudiantes; pruebas de software

Abstract: Introduction. The Final Degree Project (FDP) is a significant challenge for the completion of university studies in bachelor's degree programmes, particularly those in Engineering or Computer Science. Methods. A method was designed, consisting of defined phases and validated to identify and link specific social and professional skills related to a topic covered in a Software Engineering course and explored in depth in the FGP. The validation was carried out in a real academic context, with the participation of students who work in areas related to the subject of study. Results. The validity of the proposed methodology for identifying competencies developed during the preparation of the final degree project in relation to aspects of software testing using specific tools is demonstrated. Conclusions. The integration and deepening of technical and professional knowledge was confirmed through an experience that strengthens the competency-based approach and student-centred learning in the final stage of the degree programme. The participation of women in various roles—students, guidance counsellor, and subject teachers—is noteworthy, as it contributes to raising the visibility of their involvement in technical and academic areas of software engineering, promoting more inclusive and contextualised training practices.
Key words: Higher Education; professional skills; student-centred education; Testing software,

1. Introducción

La educación superior atraviesa una crisis de paradigma en donde las metodologías activas se presentan como una estrategia de aprendizaje significativo centrado en el estudiante.
El Proyecto Final de Carrera es un desafío para concluir los estudios universitarios en las carreras de

licenciatura. Se presenta una propuesta y su validación orientada a la profundización de temas de la Ingeniería del Software (IS) en la culminación de los estudios universitarios. en particular, la prueba o testing aplicando herramientas utilizadas en entornos profesionales sustentado en el aprendizaje centrado en el estudiante.

1.1. Metodologías activas

Las metodologías activas propician la construcción del conocimiento a partir de la resolución de problemas, el trabajo colaborativo, la toma de decisiones, el pensamiento crítico y la aplicación práctica de saberes en contextos reales. Este enfoque no solo favorece el desarrollo de competencias específicas propias de cada disciplina, sino también de competencias blandas, contribuyendo a la formación de profesionales comprometidos y con una visión integral de su desempeño.
Entre las metodologías activas más relevantes se destaca el Aprendizaje Basado en Proyectos (ABP), ampliamente abordado en la literatura reciente (Morales-Morgado et al., 2023; Araya Cortés & Sotomayor, 2025). En el ámbito de la educación superior, el ABP constituye un enfoque integrador entre teoría y práctica, ofreciendo a estudiantes y docentes una estrategia adecuada para modelar, analizar y experimentar la resolución de problemas complejos del mundo real.
En disciplinas como la Ingeniería del Software, el ABP se consolida como una metodología activa especialmente pertinente, ya que favorece el acercamiento temprano al ejercicio profesional del informático. Esta estrategia de resolución de situaciones problemáticas resulta particularmente adecuada para abordar proyectos de finalización de estudios, al promover la integración de conocimientos, habilidades técnicas y competencias transversales en escenarios auténticos.

1.2. Modelos centrados en los estudiantes

En los modelos de aprendizaje centrados en el estudiante, la formación por competencias se constituye como un marco de referencia que sustenta la educación en Informática como un proceso orientado al desempeño profesional integral (CONFEDI, 2025; Argentina, 2025; Chávez-Márquez, 2023). Este enfoque propicia el desarrollo articulado del saber, el saber hacer y el saber ser, en contextos reales y complejos, alineados con las demandas del ámbito profesional.
Desde esta perspectiva, el rol del estudiante adquiere un carácter activo, reflexivo y crítico, asumiendo la responsabilidad de participar de manera protagónica en la construcción de soluciones frente a problemáticas situadas en su contexto formativo y social (Mayorga-Ases et al., 2024; Zuñiga, 2025).
Diversos estudios respaldan la implementación de estos modelos en carreras informáticas, presentando distintas perspectivas, como aquellas que elaboran revisiones de la literatura, las que analizan las competencias (León Morejón y Gato Armas 2020; Valbuena Henao et al., 2025) y otras que se centran en las herramientas de apoyo (Battaglia et al., 2021; Glazunova et al., 2022).
En Andrade et al. (2025) se mapea el desempeño de estudiantes en temas de ingeniería del software en contextos industriales.
León Morejón y Gato Armas (2020) destacan la relevancia de formar habilidades profesionales desde el ámbito universitario, desde un espacio denominado Proyectos Informáticos, aportan a la preparación de los futuros profesionales. Valbuena Henao et al. (2025), plantean el abordaje de competencias sociales y profesionales en temas de la IS.
Battaglia et al. (2021), por ejemplo, desarrollaron un proyecto centrado en el uso de plataformas tecnológicas para el aprendizaje ubicuo colaborativo basado en competencias, aplicado a cursos vinculados con la Ingeniería de Software, particularmente durante el proceso de modelado de software. En Glazunova et al. (2022) se analizan las competencias de estudiantes de programación aplicando metodologías y herramientas ágiles.

2. Método

El método descriptivo, constó de las siguientes fases como se ilustra en la Fig. 1

Fase 1. Definición. La Fase 1, consistió en la identificación del área de conocimiento objeto de profundización del proyecto finalización de carrera. La finalidad es determinar la asignatura más afín para relevar las competencias específicas, profesionales y sociales asociadas.
Fase 2. Desarrollo, abarcó las siguientes actividades:
- Profundización de aspectos teóricos tratados en asignaturas previas.

- Determinación de elementos a contemplar en la sistematización de la experiencia centrándose en el logro de aprendizajes significativos en relación al perfil profesional
- Definición del contexto en que se desarrolla la experiencia. Se estableció el PFC como espacio de profundidad e integridad de conocimientos.
- Identificación de competencias en el proyecto final de carrera.
- Determinación de las competencias específicas y sociales y profesionales contempladas en el programa de una asignatura afín y su relación en el programa de la asignatura de finalización de estudios.

Fase 3. Validación
Con fines de validación, se optó por un proyecto de fin de carrera. En la elección se privilegió la participación de estudiantes quienes se desempeñan laboralmente en el área afín al proyecto objeto de la validación. Se propone sistematizar la experiencia e identificar las competencias específicas y sociales y profesionales contempladas en el programa de la asignatura de fin de carrera y de aquella asignatura afín.

Fig. 1- Diseño de un método orientado a validar la profundización de conocimientos de la IS en un proyecto de finalización de estudios.

3. Resultados

Los resultados de la propuesta planteada en el artículo se organizan considerando las Fases ilustradas en la Fig. 1.

3.1. Resultados asociados a las Fases 1 y 2

En referencia a las Fases 1 y 2 del método propuesto, se contextualiza la experiencia relacionando el PFC con temas tratados en la asignatura Ingeniería de Software II.
En este aspecto en particular se destacan 2 resultados de aprendizajes definidos en la asignatura Ingeniería de Software II y sus estrategias para alcanzarlos. Nótese en su alcance la articulación definida con la asignatura Proyecto Final de Carrera, el formato del Proyecto Grupal Integrador indicado, debe ajustarse al formato de presentación definido en esta asignatura.
El resultado de aprendizaje 3, se enfoca específicamente a la verificación y pruebas del sistema. Respecto al resultado 5 el mismo aporta a la formación en habilidades blandas, tan requeridas para él/la futura profesional, como la comunicación efectiva-

En el año 2025 y finalizado el cursado de la asignatura Ingeniería de Software II, se envió una encuesta anónima a los/as estudiantes con el fin de evaluar el impacto percibido sobre los resultados de aprendizaje establecidos. Los resultados fueron satisfactorios respecto a lo pretendido en cuanto a su formación profesional tanto en habilidades técnicas como blandas o actitudinales (Fig. 2). Además, se refuerza en la presentación de este trabajo en particular, aplicado en un ámbito real.

Tabla 1. Resultados de Aprendizaje (3 - 5) y estrategias para alcanzarlos (Análisis basado en Programa Analítico y de Examen para la asignatura “Ingeniería de Software II”, Res 388/23 CD)

Resultado de Aprendizaje	Unidades/ Temas	Guía de Trabajos Prácticos	Actividad Formativa	Estrategia de enseñanza
RA03 Aplica verificación y pruebas del sistema, para verificar los componentes construidos, según los requerimientos funcionales y no funcionales definidos para el proyecto. (CGT-2)	Unidad 2. Tema 7.	GTP 4 Proyecto Grupal Integrador	Clase Teórico-Práctica Clase invertida Clases Prácticas/Laboratorio	Clases teóricas y prácticas. Presenciales. Material de lectura adicional video tutoriales de implementación de normas de calidad ISO/IEC 29119. Disponible en aula virtual de complemento a la actividad presencial. Presentación de casos del mundo real, y grandes problemas ocasionados por falta de aseguramiento de calidad relacionado a verificación y pruebas. Presentación de Videos de casos reales: Toyota, Lanzamientos de naves al espacio, etc. Generar debate en clase. Aprendizaje basado en proyectos. Tutorías por UNNE Moodle.
RA05. Utiliza comunicación efectiva para realizar exposición de la solución de software diseñada utilizando conceptos, terminología específica y atributos de calidad del software, respecto a las consignas establecidas en cada actividad práctica. (CGS-1,2)	Todos los Temas del Programa Temas de investigación orientados al perfil profesional	Entregas Proyecto Grupal Integrador Entrega Guías prácticas asignadas para resolución. Entrega Tema de investigación individual	Estudio y trabajo en grupo, Estudio y trabajo individual	Defensa del Proyecto Grupal Integrador Aprendizaje basado en proyectos. Defensa de Guías prácticas asignadas para resolución. Exposición/ presentación Tema de investigación individual Presentación de los informes de cada trabajo a evaluar por UNNE Virtual

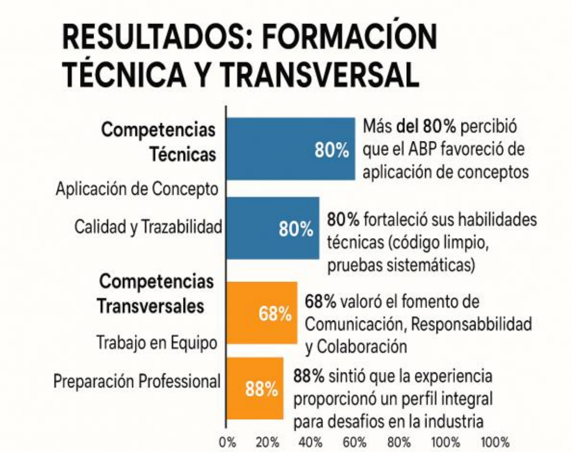


Fig 2. Resultados Encuesta alumnos/as ISII 2025.

Para validar la propuesta, se presenta el análisis de un proyecto final desarrollado por dos estudiantes, quienes fueron orientadas por una profesora que se desempeña en una asignatura de Ingeniería de Software. La coordinación del PFC, ofreció un marco general de seguimiento y monitoreo enfocándose en el desarrollo metodológico y tecnológico, a fin de cumplir el objetivo general planteado en la asignatura para concretar la finalización de la carrera. Se seleccionan como competencias definidas en la asignatura IS II que este proyecto de PFC posibilita profundizar las siguientes:

- Competencia 1: Analizar, diseñar, desarrollar, implantar, mantener y auditar sistemas de información.
- Competencia 3: Planificar, dirigir, auditar y realizar estudios de factibilidad de proyectos informáticos.
- Competencia 5: Conocer y aplicar principios éticos,

legales y profesionales que rigen la práctica profesional.

- Competencia 6: Participar activamente en grupos interdisciplinarios, adaptarse a los cambios y aprender de manera continua.
- Competencia 7: Comunicarse con efectividad en forma oral y escrita.
- Competencia 8: Utilizar técnicas, habilidades y herramientas modernas de la Ingeniería del Software.

El proyecto elegido como caso de estudio, versa sobre distintas herramientas de testing aplicadas a un desarrollo web. En particular, permite profundizar conocimientos en un área de la Ingeniería del Software en el cual las estudiantes se desempeñan laboralmente y se detallan asociados a la Fase 3: validación según el método propuesto. Cabe aclarar que esta experiencia que involucra a mujeres quienes asumen distintos roles: estudiantes, profesora orientadora, profesora coordinadora

3.2 Resultados asociados a la Fase 3. validación

En referencia a la Fase 3. Validación, se optó por el testing de software. El testing de software, realizado en paralelo al desarrollo, permite detectar y corregir problemas de forma temprana, garantizando la entrega de un producto de mayor calidad. Un software verificado y validado aporta seguridad, confiabilidad, alto rendimiento y satisfacción al cliente, además de optimizar tiempos y recursos.

El trabajo de finalización de carrera seleccionado para validar el método descrito, tiene como finalidad estudiar el proceso de pruebas de software y su aplicación práctica, integrando la elaboración de un plan de pruebas completo. Dicho plan se diseñó con el objetivo de asegurar que una aplicación cumpla con los requisitos funcionales, de rendimiento y de calidad, minimizando errores y mejorando la experiencia del usuario. Para ello, se utilizaron cuatro herramientas de pruebas automatizadas, cada una enfocada en un área específica del testing: Selenium y Cypress para la automatización de pruebas en navegadores y sitios web, JMeter para la ejecución de pruebas de carga y rendimiento, y Postman para la validación y verificación de APIs. Estas herramientas se evaluaron en una etapa práctica, lo que permitió medir su eficacia y adaptabilidad en distintos contextos.

El plan de pruebas se estructuró siguiendo estándares internacionales reconocidos, incorporando técnicas que abarcan pruebas funcionales, como unitarias, exploratorias, de integración, regresión, smoke test, aceptación y no funcionales, como las de rendimiento. La experiencia obtenida demuestra que la automatización de pruebas mejora tiempos y recursos, e incrementa la estabilidad y confiabilidad del software, constituyéndose como un elemento clave en la mejora continua del desarrollo.

El plan de pruebas propuesto se compone de cuatro fases, cada una de las cuales tiene su propio objetivo:

- Especificación: El objetivo es analizar las funcionalidades más importantes del software que se va a probar, con el fin de determinar el alcance de las pruebas y, así, establecer un cronograma de los ciclos correspondientes. Luego, si el cliente está de acuerdo, se procede con la planificación; caso contrario, se vuelve a realizar la especificación, donde se analizan los inconvenientes encontrados por el cliente, lo cual se documenta y guarda para tener una referencia para futuros proyectos.
- Planificación: El objetivo es planificar y diseñar las pruebas, ya aceptadas por el usuario. Se establecen los ciclos, las funcionalidades y el análisis de riesgo; en esta parte se genera un plan de pruebas.
- Ejecución: El objetivo es realizar las pruebas de una versión del software, y establecer las relaciones con el ciclo correspondiente.
- Evaluación y resultados: El objetivo es analizar las pruebas realizadas, la satisfacción del usuario con respecto al software probado y al proceso utilizado. La información obtenida aporta a un proceso de mejora continua. Cabe destacar que esta información es almacenada como datos históricos y se podrá utilizar para pruebas futuras.

3.2. Integración y profundización de conocimientos disciplinarios en el Proyecto Final

Se sintetizan algunos de los aspectos considerados en relación con el PFC seleccionado como caso de estudio, concebido como un espacio de profundización e integración de conocimientos. Desde el área de Ingeniería de Software y, específicamente, de la subárea de testing, dicho trabajo se articuló con:

- Modelos de desarrollo. Se utilizaron modelos de desarrollo como Scrum, adaptado específicamente para las necesidades del proyecto, junto con herramientas de control de versiones como Git y GitHub para la colaboración y gestión del código. Además, se diseñaron casos de prueba que garantizaron la cobertura de los escenarios clave. Se profundizó en la gestión ágil, ajustando la metodología Scrum al contexto del trabajo de la tesis, con una planificación detallada de Sprints y una revisión constante de los entregables para asegurar el cumplimiento de los objetivos.
- Bases de Datos. Se usaron bases de datos para almacenar datos de prueba y configurar los entornos necesarios. Además, se crearon entornos controlados que simularon distintos escenarios con datos realistas, lo que facilitó la generación de los resultados más

precisos y útiles.

- Programación. Se optó por Python como el lenguaje principal para crear los scripts de prueba. También se profundizó en el uso avanzado de librerías como Selenium WebDriver, pytest y otras herramientas que permitieron automatizar y mejorar el proceso de testing.
- Calidad del Software. Se aplicaron principios fundamentales del testing (según ISTQB -International Software Testing Qualifications Board), como: la Prueba temprana, Testing depende del contexto y Testing exhaustivo es imposible. Estos principios permiten enfocar los esfuerzos de prueba en las áreas más críticas del sistema. Se evaluaron criterios de calidad como funcionalidad, usabilidad y rendimiento, facilitando la identificación de mejoras en la experiencia del usuario y en la eficiencia del sistema. Además, se utilizaron métricas para medir y analizar los resultados. Se profundizó en la implementación de herramientas como Selenium y Allure, para automatizar pruebas, generar reportes detallados y facilitar el seguimiento de la calidad del software durante el ciclo de desarrollo.
- Metodologías de Testing. Se realizaron pruebas de regresión y pruebas de carga para evaluar distintos aspectos del software, como la estabilidad frente a cambios y el rendimiento bajo distintas condiciones. Se profundizó en la implementación práctica de estas metodologías utilizando herramientas como JMeter, para simular múltiples usuarios simultáneos y medir el tiempo de respuesta del sistema. Estas herramientas facilitaron la identificación de cuellos de botella, la automatización de pruebas repetitivas y la obtención de métricas precisas para mejorar el proceso de verificación.

3.3. Identificación de competencias desde una asignatura de Ingeniería del Software en el Proyecto Final de Carrera

A continuación, se presenta una síntesis respecto a la identificación de las competencias del perfil de egreso considerando un programa de una asignatura del área de conocimiento de la Ingeniería del Software II del plan de estudios, dado que el tema elegido para el desarrollo del PFC consiste en una profundización de conocimientos de un tema tratado en la mencionada asignatura.

Se detalla las evidencias identificadas por la profesora orientadora en el informe final del PFC respecto a las competencias especificadas.

Competencia 1: Analizar, diseñar, desarrollar, implantar, mantener y auditar sistemas de información.

Evidencia en el trabajo:

- Desarrollo de una aplicación web (“Favorite Links”).
- Diseño e implementación de pruebas automatizadas.
- Evaluación de rendimiento y funcionalidad de los sistemas.
- Uso de herramientas de testing como JMeter, Postman, Cypress y Selenium.

Competencia 3: Planificar, dirigir, auditar y realizar estudios de factibilidad de proyectos informáticos.

Evidencia en el trabajo:

- Planificación estructurada del proyecto con aplicación de Scrum.
- Análisis de factibilidad tecnológica.
- Seguimiento del ciclo de vida del proyecto.

Competencia 5: Conocer y aplicar principios éticos, legales y profesionales que rigen la práctica profesional.

Evidencia en el trabajo:

- Sección específica sobre ética profesional en el testing.
- Referencia a códigos éticos de ACM, IEEE e ISTQB.
- Discusión sobre responsabilidad del tester y gestión de información confidencial.

Competencia 6: Participar activamente en grupos interdisciplinarios, adaptarse a los cambios y aprender de manera continua.

Evidencia en el trabajo:

- Aplicación de metodologías ágiles colaborativas (Scrum).
- Reflexión sobre la mejora continua del proceso de pruebas.
- Uso de herramientas actualizadas y abiertas (aprendizaje autodidacta).

Competencia 7: Comunicarse con efectividad en forma oral y escrita.

Evidencia en el trabajo:

- Informe redactado con rigurosidad formal, clara estructuración, gráficos explicativos.
- Argumentación fundamentada en la elección de herramientas y métodos.

Competencia 8: Utilizar técnicas, habilidades y herramientas modernas de la Ingeniería del Software.

Evidencia en el trabajo:

- Uso de herramientas de testing automatizado y utilizadas en la industria.
- Aplicación de conocimientos sobre calidad de software.
- Inclusión de estándares internacionales (ISO/IEC 29119) en el diseño de pruebas funcionales y de caja negra

Se coincide con Battaglia et al. (2021) en que la competencia de los graduados, respecto a un área de conocimiento” (p. 875) (...) y la aplicación de los mismos en “futuros contextos profesionales”. El proyecto final de carrera seleccionado como caso de validación de la propuesta responde a esta afirmación, las estudiantes optaron como área de profundización de conocimientos un tema que tratan en ámbitos laborales de la industria del software aportando nuevas evidencias de estrategias superadoras en ámbitos académicos en relación a la industria del Sector de Servicios y Sistemas Informáticos.

4. Conclusiones

El artículo propone y valida un método orientado a explicitar la vinculación del Proyecto Final de Carrera con temas de Ingeniería del Software que integra aspectos del modelo centrado en competencias y en el estudiante. Su finalidad es consolidar una visión formativa coherente, actualizada y alineada con las demandas contemporáneas del campo profesional y académico de la disciplina Informática, en particular un tema de la IS, las pruebas de software integrando estándares, métodos y herramientas.

El Proyecto Final de Carrera, a partir de esta elección temática, se constituye en un espacio que integra teoría y práctica como una experiencia formativa auténtica, donde el estudiante asume un rol activo en la construcción del conocimiento, la resolución de problemas reales y la toma de decisiones fundamentadas. Esta lógica resulta especialmente pertinente en Ingeniería del Software, disciplina atravesada por la rápida evolución tecnológica, la interdisciplinariedad y la necesidad de trabajo colaborativo. Las estudiantes, quienes se gradúan con la defensa del proyecto final de carrera, manifestaron una experiencia muy valiosa, que contribuye a afianzar conocimientos en un proceso dinámico, reflexivo y sistemáticamente diseñado y validado. Espacio de profundización de aspectos clave sobre el desarrollo y el aseguramiento de la calidad de los proyectos. El proyecto de finalización de los estudios les ofreció una instancia para recorrer las etapas de su conformación y comprender los beneficios de aplicar buenas prácticas.

En particular la definición del método y su validación involucra a mujeres quienes asumen distintos roles: estudiantes, profesora orientadora, profesoras responsables de las asignaturas involucradas en la experiencia, quienes diseñan un método y optan por validar a través del testing o prueba del software. En referencia al tema seleccionado como objeto de estudio: pruebas de software, las herramientas utilizadas tienen amplia aplicación en los entornos laborales, ya que facilitan la automatización de tareas repetitivas y permiten acortar los plazos de entrega, mejorando así los resultados.

Agradecimientos

Trabajo desarrollado en el marco del Proyecto de I+D. Sistemas informáticos y Gestión del conocimiento: modelos, métodos y herramientas. Secretaria General de Ciencia y Técnica. Universidad Nacional del Nordeste, y de asignaturas de la carrera Licenciatura en Sistema de Información, Facultad de Ciencias Exactas y Naturales y Agrimensura. Universidad Nacional del Nordeste, Argentina.

Referencias

Andrade, B. A., Siqueira, R., Gomes, L. C. S., Oliveira, A., & Ribeiro, D. M. (2025). A mapping study about training in industry context in software engineering. arXiv. https://doi.org/10.48550/arXiv.2506.12590

Araya Cortés, Alexis & Sotomayor, Pedro. (2025). Metodologías Activas en la Educación Superior en Iberoamérica: Un Mapeo Sistemático. *Revista Educación Las Américas.* 14. 10.35811/rea.v14i1.330.

Argentina, Argentina.gob.ar. (2025). Capacitación docente en enfoque por competencias en ingeniería. https://www.argentina.gob.ar/noticias/capacitacion-docente-en-enfoque-por-competencias-en-ingenieria

Battaglia, N., Neil, C., De Vincenzi, M., (2021), Desarrollo y evaluación de competencias en la ingeniería de software en un entorno virtual de aprendizaje colaborativo, *XXIII Workshop de Investigadores en Ciencias de la Computación* (WICC 2021) pp 875-880.

Consejo Federal de Decanos de Ingeniería de la República Argentina (CONFEDI). (2025). Enfoque por competencias en la formación de ingenieros. https://confedi.org.ar/destacados/enfoque-por-competencias/ CO

Chávez-Márquez, I. L. (2023). Competencias digitales en universitarios a través de innovaciones educativas: una revisión de la literatura actual. *Apertura (Guadalajara, Jal.)*, *15*(2), 74-87. https://doi.org/10.32870/ap.v15n2.2398

Ferraro, M. A. Gomez Solis, L. I. (2025). De la mejora continua a la formación por competencias: Evolución en la enseñanza de Ingeniería del Software (2015–2025), *JCAS*, 7º Jornadas de Calidad de Software y Agilidad, 2025

Glazunova, O. G., Parkhomenko, O. V., Korolchuk, V. I., & Voloshyna, T. V. (2022). Building the professional competence of future programmers using methods and tools of flexible development of software applications. ICT and Learning Tools, 89(3).

León Morejón, Y. & Gato Armas, C. A. (2020). El desarrollo de habilidades profesionales para el Proceso de Software Personal. *Mendive. Revista de Educación*, 18(3), 677-688. Epub 02 de septiembre de 2020. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-76962020000300677&lng=es&tlng=es.

Mayorga-Ases, M., Tagua-Moyolema, A., Muyulema-Muyulema, D., & Velastegui-Hernández, R. (2024). Estudio sobre la implementación de metodologías activas en la educación superior: beneficios y desafíos. 593 Digital Publisher CEIT, 9(4-1), 196-208, https://doi.org/10.33386/593dp.2024.4-1.2739

Morales-Morgado, E. M., Ruiz-Torres, S., Rodero-Cilleros, S., Morales-Romo, B., & Campos-Ortuño, R. A. (2023). Metodologías activas en educación superior, mediadas por tecnologías en diversas disciplinas. *Aula*, 29, 295–311. https://doi.org/10.14201/aula202329295311

Valbuena Henao, M. A., Vallejo Córdoba, S. L., Tamayo Quintero, J. D., & Poliche, M. V. (2025). Plan estratégico para el fortalecimiento de las competencias blandas en el currículo del programa Ingeniería en Software, Informática, Ciencias Computacionales y afines. *XXVII Workshop de Investigadores en Ciencias de la Computación* (WICC).

Zuñiga, J. (2025), Metodologías activas para el desarrollo de pensamiento crítico en la universidad. Una revisión de literatura, *Espacios*, Vol. 46 (04) art. 13, DOI: 10.48082/espacios-a25v46n04p13

Computational Thinking Development: Characterization of Learning Resources

Cristiana Araújo[1], Pedro Rangel Henriques[2], João José Cerqueira[3]

ALGORITMI Research Centre/LASI, University of Minho, Braga, Portugal[1,2]
Life and Health Sciences Research Institute (ICVS), University of Minho, Braga, Portugal[3]

decristianaaraujo@hotmail.com[1], prh@di.uminho.pt[2], jcerqueira@med.uminho.pt[3]

Resumen: El Pensamiento Computacional (PC) es un proceso de razonamiento que integra un conjunto de habilidades y actitudes esenciales para resolver problemas de manera eficaz y eficiente mediante un computador. A pesar de su creciente reconocimiento en la educación básica, aún existe una brecha en la caracterización sistemática de los Recursos de Aprendizaje (RA) disponibles para su desarrollo. Este artículo propone una metodología para la categorización formal de RA basada en la ontología OntoCnE, diseñada para describir e interconectar los dominios del PC y de la Programación de Computadores. La ontología está organizada en cuatro capas que definen los dominios de conocimiento, mapean conceptos hacia los años escolares, caracterizan los RA según las habilidades de PC y la adecuación etaria, e integran las directrices de la Neuroeducación. La metodología propuesta permite construir un catálogo estructurado que apoya a los docentes en la selección de materiales para desarrollar habilidades específicas de PC, alineadas con la forma en que aprende el cerebro. Se presentan varios estudios de caso que demuestran la aplicabilidad y solidez del enfoque propuesto. Los resultados muestran que OntoCnE proporciona un marco coherente para sistematizar la catalogación de RA y fomentar prácticas docentes alineadas con el funcionamiento cerebral.
Palabras clave: Pensamiento Computacional, Ontología, OntoCnE, Recursos de Aprendizaje, Neuroeducación.

Abstract: Computational Thinking (CT) is a reasoning process that integrates a set of skills and attitudes essential for effectively and efficiently solving problems that can be executed by a computer. Despite its growing recognition in basic education, there remains a gap in the systematic characterization of the Learning Resources (LR) available for its development. This article proposes a methodology for the formal categorization of LR based on the OntoCnE ontology, designed to describe and interconnect the domains of CT and Computer Programming. The ontology is organized into four layers that define the knowledge domains, map concepts to school years, characterize LR according to CT skills and age suitability, and integrate Neuroeducation guidelines. The proposed methodology enables the construction of a structured catalog that supports teachers in selecting materials to develop specific CT skills while aligning with how the brain learns. Several case studies are presented to demonstrate the applicability and robustness of the proposed approach. The results show that the OntoCnE provides a consistent framework for systematizing the cataloging of LR and fostering brain-aligned teaching practices.

Keywords: Computational Thinking, Ontology, OntoCnE, Learning Resources, Neuroeducation.

1. Introduction

Jeannette Wing (Wing, 2006) came up with the concept of Computational Thinking (CT), in 2006, and defined it as being: *a method of solving problems, designing systems and understanding human behavior, based on fundamental concepts of Computer Science*. For Wing, CT is a crucial

approach that cuts across any field, not just for Computer Scientists. Over the years, several definitions have emerged in the literature (Aho, 2012; Sysło & Kwiatkowska, 2013; Hemmendinger, 2010; García-Peñalvo, 2018).
For us, CT is *a reasoning process characterized by a set of skills and practices (attitudes) that allow to find an effective and efficient way to solve a given problem, prone to be executed by a computer*. This definition includes, but is not limited to, the following skills: *abstraction* (identifying what is important and remove unnecessary details) (Selby & Woollard, 2013; ISTE & CSTA, 2011; Barefoot, 2021); *algorithmic thinking* (ordering a set of instructions or rules for doing something) (Selby & Woollard, 2013; ISTE & CSTA, 2011; Barefoot, 2021); *decomposition* (breaking a problem into less complex steps to reduce its difficulty) (Selby & Woollard, 2013; ISTE & CSTA, 2011; Barefoot, 2021); *patterns recognition* (identifying similarities or characteristics between problems and solving the problem using solutions previously defined in other problems and based on experiences) (ISTE & CSTA, 2011; Barefoot, 2021); *analysis* (identify and understand the different elements of a problem) (Abraham, 2005; Dalal, Carberry, & Archambault, 2021); *logical reasoning* (deduction (conclude something from information that already exists), induction (affirm a generalized truth based on the observation of some elements), abduction (look for new ideas and knowledge that can validate something)) (ISTE & CSTA, 2011; Dolgopolovas, Dagiené, Jasuté, & Jevsikova, 2019; Barefoot, 2021); and *evaluation* (ensure that the solution is adequate to solve the problem, checking the limit cases) (ISTE & CSTA, 2011; Selby & Woollard, 2013; Barefoot, 2021).

These skills are supported and enhanced by practices or attitudes, namely: *adapting* (adjust the strategy/resolution to each situation) (Abraham, 2005; Dalal et al., 2021); *collaborating* (working as a team) (ISTE & CSTA, 2011; Barefoot, 2021); *creating* (design and make with creativity) (Barefoot, 2021); *debugging* (finding and fixing errors) (Barefoot, 2021); *optimizing* (restructure for a more efficient solution) (Abraham, 2005; Dalal et al., 2021); *persevering* (continue and never give up, even when the problem is more complex to solve) (ISTE & CSTA, 2011; Barefoot, 2021); *planning* (define long-term objectives and the means to achieve them) (Abraham, 2005; Dalal et al., 2021); and *tinkering* (change and see what happens - cause and effect experiments) (Barefoot, 2021).

To develop CT skills from a young age, it is necessary to use appropriate Learning Resources (LR). There are many LR available, online and in books, to promote CT. However, in addition to being dispersed, it is also difficult to quickly understand the CT skills that each of them develops, and the scholar years that the LR can be applied. In this article, we aim to present an approach to categorize these LRs. This categorization will help the teacher to more quickly and efficiently find an LR to train one or more specific CT skills for a given school year.
Furthermore, this categorization will make it possible to build an LR catalog in order to concentrate all LR in one place. To implement the LR categorization approach, we will use the OntoCnE Ontology. We started building OntoCnE some time ago with the aim of understanding how to promote the development of CT in children. We believe that developing CT skills from an early age can help students reduce the difficulties they have in learning to solve problems using a computer, thus reducing the known failure rate in Computer Programming (CP) courses (Gomes & Mendes, 2007; Lawan, Abdi, Abuhassan, & Khalid, 2019). As will be detailed in the next section, OntoCnE is composed of 4 layers. Layers 3 and 4 are crucial to perform LR categorization. Layer 3 characterizes LR in order to select them for concrete training (example, development of *Pattern Recognition* in *4th year*). Layer 4 assists the teacher in selecting a concrete LR that aids training based on one or more specific Neuroeducation Guidelines (e.g., developing *Pattern Recognition* in *4th grade* with *Increased Cognitive Load*). This categorization allows the teacher to select the most appropriate and efficient LR in developing one or more CT skills.

This article is structured into four main sections. Section 2 presents the OntoCnE ontology and its four-layer structure, explaining how it supports the formal categorization of Learning Resources (LR). Section 3 illustrates the proposed methodology through several case studies that demonstrate the applicability and robustness of the approach. Finally, Section 4 discusses the main conclusions and outlines directions for future research.

2.OntoCnE, the backbone for characterization

As previously mentioned, we believe that CT developed from a young age can help students improve their problem-solving abilities. These skills will be crucial for everyone, especially for those entering Computer Programming (CP) Courses. To understand how the CT and CP domains relate, we decided years ago to create an ontology to clarify and formalize this relationship.

This ontology, in its genesis, was built by a multidisciplinary team (experienced University CP Professors, IT and other areas K-12 teachers), and over the years it has continued to be validated by several IT teachers from K-12. Recently, we contrasted this ontology with the *Reference curriculum in technology and computing: from early childhood education to primary education* from Brazil (Raabe, Brackmann, & Campos, 2018), and *Essential Learning of Information and Communication Technologies (TIC–Tecnologias da Informação e Comunicação) – Basic Education*[1] from Portugal.

An Ontology describes a domain of knowledge, identifying:domain *concepts* and *relationships* among concepts. Concepts are characterized by zero, one or more *attributes*. An ontology also includes *individuals* who are instances of concepts. In addition to concepts and relationships, the ontology is composed of several *triples*. A triple has three elements – *Subject, Predicate, Object* – where *subject* and *object* are concepts or individuals, and *predicate* is a relation (Araújo, Henriques, & Cerqueira, 2023b).

OntoCnE is composed of 4 layers that: describe CT and relate it with CP allowing an understanding of how and why to develop CT (Layer 1); define which concepts should be worked at each scholar year and how deep to introduce them (Layer 2); define the skills developed by each resource and the appropriate ages (Layer 3) (Araújo et al., 2023b); describe a set of Neuroeducation Guidelines relating them with the learning process and linking with the resources (Layer 4).

OntoCnE is formally written in our DSL called OntoDL+, which was specifically designed to facilitate the process of defining and instantiating an ontology. For interoperability purposes, the OntoDL+ compiler generates OWL.

2.1. Layer 1: Learning Domain Description

Layer 1 of OntoCnE aggregates a set of concepts that describe the CT and CP domains, such as *Abstraction, Algorithm, Automaton, Binary System, Decomposition, Debug, Encryption, Evaluation, Pattern Recognition, Planning, Problem, Program, Regular Expression, Tinkering, Variable*, among others.

After defining the concepts, it was necessary to identify a set of relationships that connect the concepts to give meaning to the universe of discourse. Some of the relationships we gathered were: *acts on, can be, detects, guides, is composed of, solves, requires*, among others.

After identifying concepts and relations, the necessary triples (*Subject=Predicate=>Object*) were defined. In Figure 1 we can see a fragment of this OntoCnE component that describes two main concepts: *Algorithm* and *Computational Thinking.*

```
Triples{
Algorithm =[
  requires => Computational_Thinking;
  is_realized_by => Program;
  is_composed_of => Instruction;
  performs => Decryption, Encryption];
Computational_Thinking =[
  involves => Adaptation, Collaboration, Creation,
     Debugging, Optimization, Perseverance,Planning,
     Tinkering;
  requires => Abstraction, Algorithmic_Thinking,
     Analysis, Decomposition, Evaluation,
     Logical_Reasoning, Pattern_Recognition ]; ...}.
```

Figure 1. Ontology triples that describe Algorithm and Computational Thinking.

By reading the triples it is possible to see easily how the concepts relate to each other to form an understandable and meaningful description. From Figure 1 it is possible to conclude facts such as: an *Algorithm* requires *Computational Thinking* and is composed of *Instruction* (one or more). *Computational Thinking* involves *Tinkering* and

[1] *Accessible in:*https://www.dge.mec.pt/aprendizagens-essenciais-ensino-basico

requires *Decomposition*. Figure 2 provides information about the description of the CT and CP domains and how they relate to each other.

```
Triples{
Debugging =[ analyzes => Program;
             requires => Perseverance ];
Tinkering =[ involves => Exploration ];
Exploration =[ involves => Evaluation, Test ];
Program =[ is_composed_of => Code_Block;
           is_guided_by => Automaton;
           is_written_in => Language;
           requires => Test;
           solves => Problem ];
Code_Block =[ is_composed_of => Instruction ];
Automaton =[ guides => Program;
             is_equivalent_to =>Regular_Expression];
Regular_Expression =[
        defines => Pattern;
        is_used_in => Pattern_Recognition ];...}
```

Figure 2. Ontology triples that link CP and CT domains (fragment).

From Figure 2 it is possible to infer that *Tinkering* involves *Exploration*. And *Exploration* involves *Evaluation* and *Test* the solution. An *Automaton*, in addition to guiding a *Program*, is equivalent to a *Regular Expression*. *Regular Expression* defines a pattern and therefore it can be said that it is used in *Pattern Recognition*. Figure 3 displays a fragment of OntoCnE Layer 1.

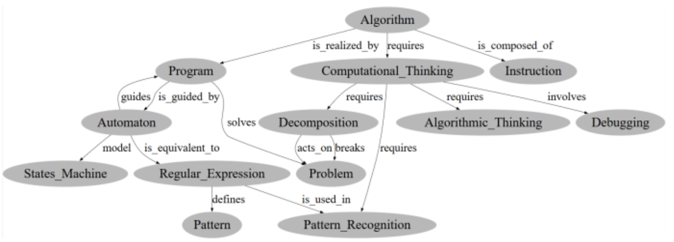


Figure 3. OntoCnE – Layer 1 (fragment).

2.2. Layer 2: Mapping to School Years

Layer 2 of OntoCnE defines what to work on at each teaching level and the depth level. In this layer we added one concept –*Year*– that represents a scholar year. We added nine individuals (*Year1, Year2, Year3, Year4, Year5, Year6, Year7, Year8, Year9*) to instantiate the concept *Year*. These individuals represent each scholar year, that compose the official cycles of basic education (1st, 2nd and 3rd cycles) in Portugal.

In addition to this concept and individuals, we also added thirteen new relationships, namely: *uses, knows, applies, reinforces, optimizes, identifies, creates, fixes, executes, modifies, examines, fosters, develops*. These relationships are essential for this layer because they reflect the degree of conceptual elaboration that students will learn throughout each scholar year.

We assign a meaning to each relationship to accurately define the degree of strengthening of a theme throughout a given year:

- *uses* – the student resorts, without formalization or explanation, to a concept to be taught later;
- *knows* – the student becomes aware of a concept that is formally defined for the first time;
- *applies* – student resorts to the use of a concept, consciously and autonomously, after having been introduced to it in a previous year;
- *reinforces* – the student remembers and reuses a concept and improves his knowledge by adding complexity;
- *examines* – the student analyzes in detail and understands a concept;
- *optimizes* – the student analyzes something he has built and improves or perfects the initial solution;
- *identifies* – the student analyzes something he has built and finds an anomaly or behavior;
- *creates* – the stud builds a new object/artifact from scratch;
- *fixes* – the student analyzes something he has built and solves an anomaly or behavior, which he previously identified;
- *executes* – the student performs an action, such as running a program;
- *modifies* – the student changes an object to evaluate the consequence of the modification introduced;
- *develops* – the student acquires/deepens a skill;
- *fosters* – the student acquires/deepens a practice/attitude.

These relations allow the linkage of each individual added in Layer 2, with the concepts of Layer 1. Figure 4 shows a snippet of the triples created in Layer 2.

Observing the triples described in Figure 4 the student will use the concept *Program* without formalization in *Year1*, and in *Year2* he will formally learn its definition. On the other hand, the concepts *Algorithm* and *Instruction* are formally presented in *Year1*. In the following year, *Year2*, the student will reinforce these concepts, remembering and adding more details about them. The relationships that affect concepts such as *Problem*, *Decomposition*, *Error*, *Collaboration* are the same over the years. However, the student's skills, over the years, will evolve and he will be able to examine more complex problems and divide them into smaller parts with greater ease. Furthermore, he will also be able to identify and correct deeper errors, and he will a work in a team more easily.

```
Triples{
Year1 =[ iof => Year];
Year2 =[ iof => Year];
Year1 =[
  examines => Algorithm, Problem;
  creates => Algorithm;
  uses => Internet, Program, Visual_Lang;
  knows => Algorithm, Computer, Instruction;
  identifies => Error;
  fixes => Error;
  develops => Algorithmic_Thinking, Decomposition,
    Pattern_Recognition;
  fosters => Collaboration, Debugging ...];
Year2 =[
  examines => Algorithm, Problem;
  creates => Algorithm;
  uses => Programming_Environment, Variable;
  knows => Atomic_Type, Code_Block, Internet,
    Program, Repetition, Visual_Lang;
  reinforces => Algorithm, Instruction;
  executes => Program;
  applies => Computer;
  identifies => Error;
  fixes => Error;
  develops => Algorithmic_Thinking, Decomposition,
    Pattern_Recognition, Evaluation;
  fosters => Collaboration, Debugging, Perseverance,
     Planning ... ]; ...}.
```

Figure 4. Ontology triples that describe what to work at each scholar year – Layer 2.

Figure 5 displays a fragment of OntoCnE connecting Layers 1 and 2 illustrating the main idea of our work. In this figure, Layer 1 is represented in turquoise and Layer 2 in red. Concepts are represented in ellipses as in the previous layer and individuals are represented in rectangles.

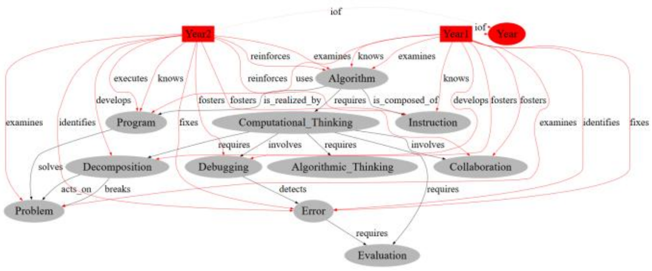


Figure 5. OntoCnE – Layer 1 and Layer 2 (fragment).

2.3.Layer 3: Learning Resources Characterization

Layer 3 of OntoCnE aims to characterize each Learning Resource (LR) by defining the skills developed by the resource and the years in which it can be worked. So in Layer 3 we added a new concept – *Learning Resource* – with several attributes that characterize the artifact, namely: *Title* (LR name); *Learning Objective* (concepts/skills that LR develops); *Description* (text that explains what the LR is and the challenges/tasks it contains); *Category* (eg: game, worksheet, video, etc.); *Type* (unplugged, plugged or both); *Material* (what is needed to use the LR - eg: paper and pencil, computer, robot, etc.); *Usage Rules* (guidelines for using the LR); *Evaluation Method* (LR assessment guidelines); *Age* (age at which it should be used); *Duration* (average time necessary to complete the activity); *Subjects Involved* (topics that LR addresses - eg: Math, History, English); *Language* (language to be used to perform/complete the activity proposed by the LR); *Requirements* (background necessary to execute that activity); *Author* (person who designed the LR). We have identified carefully those attributes in order to create a model that gathers the main characteristics of a LR, so that a teacher can select appropriately a LR to develop certain skills in a given school year.

In Layer 3 we also add individuals (instances of the *Learning Resource* concept). To explain how a LR instance is included in the ontology the generic *LR Example* individual is added in the OntoCnE fragment shown in Figure 6.
The *Learning Resource* concept will be instantiated with concrete resources that are the real artifacts that will be used to develop CT skills (*LR Example* denotes a possible instance). In Layer 3 we also added two new relations: *train* (relates each

individual *Learning Resource* to the concepts it develops); and *resort* (relates the concept *Year*, from Layer 2, with the concept *Learning Resource* and respective individuals).

In Figure 6 we can see a fragment of Layer 3 triples which describes a concrete LR and establishes the concepts it works (concepts that are defined in Layer 1) and the school years in which it can be used (Layer 2). Ontology triples describing each learning resource and the school years where it applies.

```
Triples{
 LR_Example=[ iof=> Learning_Resource[title='Z'...];
             trains=> Decomposition, Algorithm...];
 Year =[ resorts=> Learning_Resource ];
 Year2 =[ iof => Year;
         resorts=> LR_Example... ];  ...}.
```

Figure 6. Ontology triples that characterize each LR and the school years where it applies.

Figure 7 shows a fragment of the 3 layers of OntoCnE. Layer 3 is represented in orange and describes the triples presented in Figure 6.

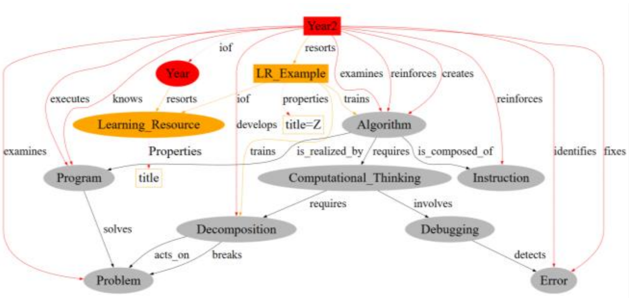


Figure 7. OntoCnE – Layer 1, Layer 2, and Layer 3 (fragment).

2.4. Layer 4: Neuroeducation Integration

Layer 4 of OntoCnE aims to describe the domain of Neuroeducation Guidelines and complement the characterization of each LR. This layer was built to formalize the choice and adaptation of a set of pedagogical Guidelines based on Neuroeducation evidences, which can be consulted in detail in (Araújo, Henriques, & Cerqueira, 2023a). For that purpose, we added 42 new concepts, such as: *Cognitive Closure, Collaborative Work, Concept Map, Connection Concepts, Diversified Repetition, Icebreaker, Increasing Cognitive Load, Knowledge Integration, Multisensory Stimulation, Student, Spaced Repetition, Strong Emotion, Student Choice, Teacher*, among others. In addition to the concepts, we also identified 11 new relationships, namely: *does, experiences, improves, increases, is interconnected, feels, monitors, must follow, potentiates, promotes, produces*. Figure 8 shows the characterization of Guideline 2 – *Progressively increasing Cognitive Load benefits learning*.

```
Triples{
 Increasing_Cognitive_Load =[
    is-a=>Neuroeducation_Guideline];
 Increasing_Cognitive_Load =[
    must_follow => Student_Progress;
    improves => Learning;
    increases => Intrinsic_Motivation ];
 Learning_Resource =[ promotes =>
    Increasing_Cognitive_Load ];
 Teacher =[ promotes => Increasing_Cognitive_Load ];
 Student =[
    experiences => Increasing_Cognitive_Load]; }
```

Figure 8. Ontology triples to characterize Guideline 2.

Reading Figure 8 it is possible to infer that: *Increasing Cognitive Load* must be in accordance with *Student Progress* to enhance his *Intrinsic Motivation*. *Increasing Cognitive Load* can be stimulated by *Learning Resource* or *teacher*. In Figure 9 presents the characterization of Guideline 8 – *Cognitive Closure enhances learning and memory consolidation*.

```
Triples{
 Cognitive_Closure =[
   is-a=>Neuroeducation_Guideline];
 Cognitive_Closure =[
   improves => Learning, Long_Term_Memory_Storage;
   powers => Recovery_Information ];
 Recovery_Information =[
   powers => Long_Term_Memory_Storage ];
 Comment_Solution =[ powers => Cognitive_Closure ];
 Teacher =[ promotes => Cognitive_Closure ];
 Student =[ does => Cognitive_Closure ]; }
```

Figure 9. Ontology triples to characterize Guideline 8.

Cognitive Closure in addition to improving *Long Term Memory Storage*, also powers *Recovery Information*. *Cognitive Closure* can be promoted by the teacher or powers through *Comment Solution* of a problem.

After characterizing the Neuroeducation Guidelines, it is necessary to characterize the Learning Resources under the light of those Neuroscience-based orientations. To illustrate how we characterize an LR in Layer 4, we will again use the individual *LR Example* (see Figure 10).

```
Triples{
 LR_Example =[ promotes => Cognitive_Closure,
                         Increasing_Cognitive_Load,
                         Multisensory_Stimulation ]; ...}
```

Figure 10. Ontology triples to characterize LR according to Neuroeducation Guidelines.

Reading Figure 10 we can infer that *LR Example* promotes *Cognitive Closure*, *Multisensory Stimulation*, and *Increasing Cognitive Load*. Figure 11 shows a fragment of OntoCnE with the 4 Layers connected. Layer 4 is represented in green.

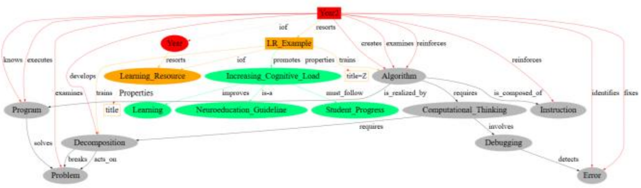


Figure 11. OntoCnE – Layer 1, Layer 2, Layer 3, and Layer 4 (fragment).

Layer 4 clearly supports the teacher in selecting a concrete LR that promotes the development of CT skills based on one or more specific Neuroeducation Guidelines. To view in detail the 4 layers of OntoCnE, consult the web platform: https://computationalthinking4all.epl.di.uminho.pt/ontoCnE.html.

3. Learning Resources formal cataloguing

OntoCnE, in addition to describing the concepts to be worked on, in each school year, and their level of in-depth study, is also useful for characterizing Learning Resources (LR). This characterization of LRs makes possible to select them for a more effective concrete training. To illustrate the characterization of LR in OntoCnE, we will use 4 concrete LRs (built by our of *Master's in Informatics Teaching* students) as case studies.

3.1. Case Study 1: Let's catch all the Pokémon

The first case study presents *Let's catch all the Pokémon*, which for simplicity we will just call *Pokémon*. This LR is a short book with a story based on Nintendo's *The Pokémon Company* animated series. The objective of the story is to help *Ash Ketchum*, the main character, to become a Pokémon Master. For that purpose, the student needs to help Ash Ketchum. Throughout the story, 3 challenges arise for the student to help Ash fulfill his mission.

In the first challenge, the student receives the image of a Pokémon on the gridded table and the legend with the color that represents each number. Student must analyze the table, line by line, and record for each color that appears the number of squares of the color and then the color code (eg:*2-3*, that is, *2 squares in gray (color 3)*). The challenge outcome must be is the encoding of the Pokémon image (see Figure 12). The second challenge aims to do exactly the opposite of the first one that is, student receives the encoding of the image and must paint it on the gridded table to discover the Pokémon. Finally, in the third challenge, similar to the first, student must paint the gridded table according to the color code. However, this time, the color code corresponds to a multiple of a number (eg. numbers that are multiples of 3 are colored white).

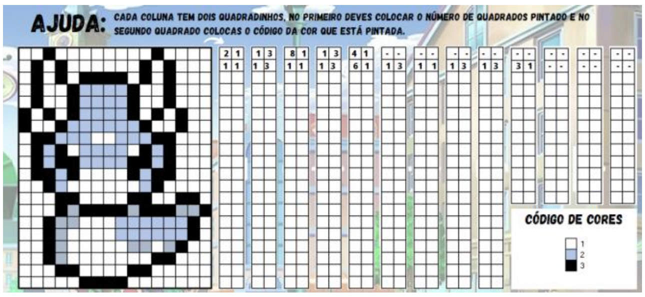


Figure 12. First Pokémon challenge: encoding the figure.

This LR develops the following CT skills: Analysis (challenges 1-3); Algorithmic Thinking (challenges 1-3); Pattern Recognition (challenges 1-3); Logical Reasoning (challenge 3). In addition to these skills, this LR also works CP concepts, namely: Repetition (challenges 1 and 2); Encryption and Decryption (challenges 1-3). In Figure 13 a fragment of triples is presented that shows how Layer 3 characterizes the LR Pokémon and relates it to Layers 1 and 2. Furthermore, it also demonstrates how Layer 4 relates to Layer 3.

Figure 13 identifies Pokémon as a concrete LR (line 1 and 6); characterizes it through attributes (lines 7-19); identifies Year 4 as the appropriate level for its use (4-5); describes the skills it develops (20-22); and describes the Neuroeducation Guidelines that the concrete LR Pokémon is in accordance (23-24).

```
1  Individuals{ Pokemon }
2  Triples{
3   Year =[ resorts=> Learning_Resource ];
4   Year4 =[ iof => Year;
5            resorts=> Pokemon ];
6   Pokemon =[ iof => Learning_Resource[
7    title = 'Let's catch all the Pokemon',
8    learningObjective='LR aims to develop Logical...',
9    description='LR contains 3 challenges...',
10   category = book,
11   type = unplugged,
12   material='book and colored pencils',
13   usageRules = 'see manual annex',
14   evaluationMethod='conclude each activity with...',
15   age = '9+',
16   duration = '100 minutes',
17   subjectsInvolved = Mathematics, IT,
18   language = 'Portuguese',
19   author = 'Ines Cancela' ]  ];
20  Pokemon =[ trains=> Analysis, Logical_Reasoning,
21   Algorithmic_Thinking, Pattern_Recognition,
22   Repetition, Encryption, Decryption...];
23  Pokemon =[ promotes => Increasing_Cognitive_Load,
24   Diversified_Repetition, Cognitive Closure ]; ...}.
```

Figure 13. Ontology triples to characterize Pokémon as a LR for Year 4.

3.2. Case Study 2: Mática Escape Room

The second case study presents an Escape Room called *Mática*. This Escape Room is about the Mática Engineer who is in a Secret Computer Base, but after the infiltration of the Malicious Virus, all the doors are blocked, and a bomb threatens to explode in 50 minutes. To unlock the doors and save Mática, the student must successfully solve 5 challenges about Regular Expressions or Automaton. Each challenge has more than one answer options, but only one is correct.

The first challenge presents a Regular Expression (RE) and the student has to discover, from it, which programming language Mática is using. In the second challenge, Mática needs a numeric code. The student has to select the correct numeric code based on the RE provided. In the third challenge, the player must select the correct email, from the RE provided, to unlock the door. In the fourth challenge, the student has to help Mática select the shortest path. In this challenge, an Automaton is presented, and the student has to select the option that contains the shortest path in the Automaton (see Figure 14). In the fifth challenge, Mática found a message encoded in ASCII code where each number represents a specific character (eg: a-97). The student, after analyzing the RE provided, must select the correct character sequence that matches the RE. It is important to highlight that to solve this LR, students must have knowledge of RE and Automaton. The Escape Room is available at: https://bit.ly/matica-escape-game.

Figure 14. Fourth challenge of Mática: selecting the shortest path.

This LR, throughout the 5 challenges, promotes the development of the following CT skills: Analysis; Pattern Recognition; Logical Reasoning; and Evaluation. Furthermore, it also allows to work on CP concepts, namely: Regular Expressions (challenges 1-3, 5); Automaton (challenge 4); Encryption and Decryption (challenges 1-3, 5). Figure 15 presents a fragment of triples that characterize LR Mática.

```
Individuals{ Matica }
Triples{
 Year =[ resorts=> Learning_Resource ];
 Year9 =[ iof => Year;
          resorts=> Matica ];
 Matica =[ iof => Learning_Resource[
   title = 'Matica Escape Game',
   learningObjective='Aims develop Pattern
    Recognition...',
   description='LR contains 5 challenges...',
   category = game,
   type = Plugged,
   material='computer',
   usageRules = 'see manual annex',
   evaluationMethod='complete the Escape Room...',
   age = '14+'),
   duration = '50 minutes',
   subjectsInvolved = Computer Programming,
   language = 'Portuguese',
   author = 'Claudia Ribeiro, Emanuel Castro and
    Silvia Barros' ] ];
 Matica =[ trains => Analysis, Pattern_Recognition,
  Logical_Reasoning, Evaluation, Automaton,
  Regular_Expression, Encryption, Decryption  ...];
 Matica =[ promotes => Increasing_Cognitive_Load,
  Diversified_Repetition, Collaborative_Work]; ...}.
```

Figure 15. Ontology triples to characterize Mática as a LR for Year 9.

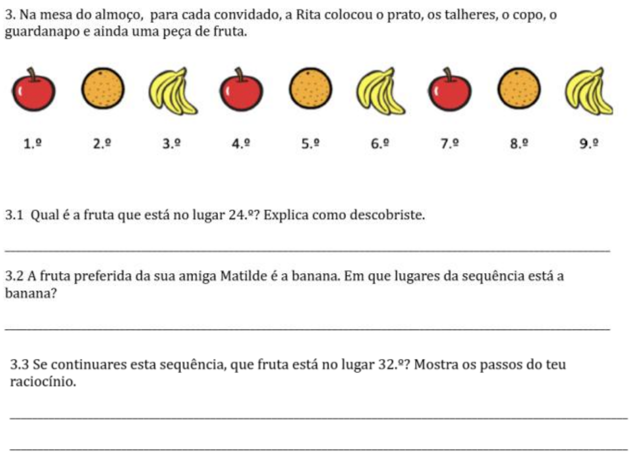
Mática is a concrete LR (line 1 and 6); is characterized through its attributes (lines 7-20); must be used in Year 9 (4-5); promotes the development of CT and CP skills, such as *Pattern Recognition* and *Regular Expression* (21-22); and promotes mechanisms that enable the application of Neuroeducation Guidelines, such as *Diversified Repetition* and *Collaborative Work* (24-25).

3.3. Case Study 3: Rita's party – Discover the hidden patterns!

The third case study presents *Rita's Party – Discover the hidden patterns!,* a paper-based and unplugged LR designed for primary school students. The activity is organized into four progressive challenges embedded in a playful narrative about a girl named Rita preparing her birthday party.

In the first challenge (*The Rabbit's Jumps*) **s**tudents are introduced to a sequence describing a rabbit's alternating jumps — small, large, small, large, and so on. They are asked to predict the type of jump at specific positions (e.g., 7th and 20th) and to describe the underlying rule. In the second challenge (*The Balloons*), students analyze a repeating sequence of balloon colors (blue, red, green, blue, red, green...) and determine the color at various positions and identify pattern of the balloon sequence. In the third challenge (*The Fruits on the Table*). Here, each guest's place at the table includes a specific fruit in a fixed order. Students must infer the pattern to predict which fruit appears at particular positions (e.g., 24th or 32nd) and identify where a given fruit (banana) occurs in the sequence. In this challenge, students must present the steps of their reasoning for each question (see
Figure 16). Finally, in the fourth challenge (*The Color Code*), students decode a grid following a symbolic color legend (B – White, V – Red, A – Yellow), paint the squares accordingly, and identify visual patterns that emerge in the completed figure. Afterwards, they are asked to represent the same sequence in a more efficient way, such as *2B 1V 1B 1V 3B*,

Throughout the four challenges, this LR promotes the development of several CT skills: Pattern Recognition (challenges 1–4), Algorithmic Thinking (challenges 1–4), Analysis (challenges 1–4), and Logical Reasoning (all challenges). Furthermore, it encourages the development of the Optimization attitude, as students are invited to find shorter and more efficient ways to represent information (challenge 4).


3. Na mesa do almoço, para cada convidado, a Rita colocou o prato, os talheres, o copo, o guardanapo e ainda uma peça de fruta.

3.1 Qual é a fruta que está no lugar 24.º? Explica como descobriste.

3.2 A fruta preferida da sua amiga Matilde é a banana. Em que lugares da sequência está a banana?

3.3 Se continuares esta sequência, que fruta está no lugar 32.º? Mostra os passos do teu raciocínio.

Figure 16. Third challenge of Rita's party: predicting which fruit appears in certain positions.

Figure 17 presents a fragment of triples that characterize Rita's party LR.

```
1  Individuals{ RitaParty }
2  Triples{
3   Year =[ resorts=> Learning_Resource ];
4   Year3 =[ iof => Year;
5            resorts=> RitaParty ];
6   RitaParty =[ iof => Learning_Resource[
7     title = 'Rita's party - Discover the hidden
       patterns!',
8     learningObjective='Aims develop Pattern
       Recognition...',
9     description='LR contains 4 challenges...',
10    category = Worksheet,
11    type = Unplugged,
12    material='worksheet',
13    usageRules = 'see manual annex',
14    evaluationMethod='complete the worksheet...',
15    age = '8+'),
16    duration = '50 minutes',
17    subjectsInvolved = Maths,
18    language = 'Portuguese',
19    author = 'Cristiana Araujo' ] ];
20  RitaParty =[ trains => Pattern_Recognition,
21  Algorithmic_Thinking, Analysis, Logical_Reasoning,
       Optimization ...];
22  RitaParty =[ promotes => Increasing_Cognitive_Load,
23   Diversified_Repetition ]; ...}.
```

Figure 17. Rita's party – Discover the hidden patterns! as a LR for Years 3 and 4.

According to OntoCnE Layer 3, Rita's party is a concrete LR (line 1 and 6); is characterized through its attributes (lines 7-19); is recommended for Year 3 and Year 4 (4-5), as it provides a concrete and narrative-driven environment that facilitates early CT skill development. This LR promotes the development of CT and CP skills, such as *Pattern Recognition, Algorithmic Thinking*, *Analysis*, *Logical Reasoning,* and *Optimization* (20-21); and promotes mechanisms that enable the application of Neuroeducation Guidelines, such as *Increasing Cognitive Load*, and *Diversified Repetition* (22-23).

3.4. Case Study 4: LOST GPS

The fourth case study presents *LOST GPS*, a digital and interactive LR developed in Canva, designed to teach Regular Expressions (RE) and Automata theory in a playful and exploratory way. The resource features an engaging narrative that guides the student throughout the game, promoting immersion and contextualization of the tasks. The activity takes place in a fictional city where all street signs are written using regular expressions. The student's goal is to help João, whose GPS is malfunctioning, reach a restaurant by correctly interpreting the street names based on the RE displayed by the device.

Throughout the game, students face five progressive challenges, each corresponding to a city intersection. At each crossroads, several street signs are displayed, but only one matches the regular expression pattern provided by the GPS. Students must analyze the syntax of each expression, identify the correct pattern, and decide which path to follow. As the journey progresses, the patterns become gradually more complex, introducing key RE concepts such as alternation (|), optionality (?), repetition (* and +), and character ranges ([a-z]) (see Figure 18). After completing the route, students carry out two complementary tasks. In the first, they reconstruct the word formed by the initials of the streets correctly visited. In the second, they design a finite automaton representing the entire route, identifying states, transitions, and inputs corresponding to each movement in the game. The LOST GPS LR is available at: https://bit.ly/lostGPS.

Throughout all challenges, this LR fosters the development of key CT skills, including Analysis, Pattern Recognition, Algorithmic Thinking, Logical Reasoning, and Evaluation, which are consistently reinforced across the different tasks. Furthermore, it also allows to work on CP concepts, namely: Regular Expressions (in the first 5 challenges); Automaton (in the 2nd supplementary challenge); Decryption (in the first 5 challenges).

Figure 18. First challenge of LOST GPS: Regular expression with alternation and optionality.

Figure 19 presents a fragment of triples that characterize LOST GPS LR.

```
1  Individuals{ LostGPS }
2  Triples{
3   Year =[ resorts=> Learning_Resource ];
4   Year9 =[ iof => Year;
5          resorts=> LostGPS ];
6   LostGPS =[ iof => Learning_Resource[
7     title = 'LOST GPS',
8     learningObjective='Aims develop Pattern
       Recognition...',
9     description='LR contains 5 challenges and 2
       complementary challenges...',
10    category = Game,
11    type = Plugged,
12    material='Computer, internet and notebook and
       pencil',
13    usageRules = 'see manual annex',
14    evaluationMethod='complete the game...',
15    age = '15+'),
16    duration = '50 minutes',
17    subjectsInvolved = Mathematics, IT,
18    language = 'Portuguese',
19    author = 'Diogo Fernandes e Rui Cardoso' ] ];
20  LostGPS =[ trains => Pattern_Recognition,
       Algorithmic_Thinking, Analysis,
       Logical_Reasoning, Evaluation,
       Regular_Expression, Automaton, Decryption...];
21  LostGPS =[ promotes => Increasing_Cognitive_Load ];
       ...}.
```

Figure 19. LOST GPS as a LR starting in the Year 9.

According to OntoCnE Layer 3, LOST GPS is a concrete LR (lines 1 and 7), characterized by its attributes (lines 8–19). It is recommended for Year 9 (lines 3–4), as it provides a digital, game-based environment that supports the acquisition of advanced CT skills. This resource promotes the development of several competencies, namely

Pattern Recognition, Algorithmic Thinking, Analysis, Logical Reasoning, Evaluation, and *Decryption* (lines 20–21), and introduces fundamental Computer Programming concepts such as *Regular Expressions* and *Automata*. Furthermore, it activates mechanisms that enable the application of Neuroeducation Guidelines, particularly *Increasing Cognitive Load* (line 22), through the gradual introduction of progressively more complex challenges throughout the game.

The characterization of LRs based on OntoCnE clearly supports the teacher in selecting a concrete LR that promotes the development of CT and CP skills based on one or more specific Neuroeducation Guidelines (e.g., developing *Pattern Recognition* in *9th year* with *Diversified Repetition*). It is important to highlight that, even if a LR does not directly include a guideline, the teacher can look for pedagogical strategy to promote it. For example, Mática does not include the *Cognitive Closure* Guideline, but the teacher can ask students to identify the strategy they used to solve the exercises. It is important that everyone recognizes that the main strategy used was *Pattern Recognition*. Furthermore, to encourage collaboration, the teacher will also include the *Collaborative Work* Guideline and form groups of 2 or 3 students to solve the Escape Room.

4. Conclusion

The development of Computational Thinking (CT) from the early years of schooling is essential to form citizens capable of understanding and solving problems in a structured and efficient way. The OntoCnE ontology constitutes a conceptual model that organizes and relates the main concepts of CT and Computer Programming with Learning Resources (LR) and Neuroeducation evidence, enabling a more informed and effective pedagogical application.

The methodology presented for the formal categorization of LR demonstrates that it is possible to build a structured catalog that supports teachers in the careful selection of materials suitable for each skill and school level. The application of OntoCnE contributes to a more structured and evidence-based pedagogical practice by supporting the selection of LR aligned with neuroeducational principles and with the progressive development of CT skills. Several case studies were analyzed to illustrate the applicability and robustness of the proposed characterization approach.

As future work, it is intended to expand the number of characterized LR and to integrate the ontology into a digital repository that allows queries, filtering by skills and attitudes of CT and Neuroeducation Guidelines, as well as active collaboration between teachers and researchers. Furthermore, empirical studies will be conducted in real educational contexts to assess the impact of using the OntoCnE catalog on the development of students' CT skills. Finally, the progressive integration of OntoCnE into teacher training programs is foreseen, promoting pedagogical practices grounded in Neuroeducation evidence. This integration began in the current academic year, within the *Master's in Informatics Teaching* at University of Minho, where OntoCnE was formally included in the syllabus of a course unit in the newly revised edition of the program.

Acknowledgment

This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UID/00319/2023.
The Ph.D. work of Cristiana Araújo is supported by FCT – Fundação para a Ciência e Tecnologia, Research Grant, with reference 2020.09845.BD, *https://doi.org/10.54499/2020.09845.BD*.
We are in debt to all our students of *Master's in Informatics Teaching* (eds. 22-23, 23-24, and 24-25) for having developed the Learning Resources that made possible this article.

References

Abraham, S. (2005). Stretching strategic thinking. Strategy & leadership, 33(5), 5–12.

Aho, A. V. (2012). Computation and Computational Thinking. *The Computer Journal*, 55(7), 832-835. doi: https://doi.org/10.1093/comjnl/bxs074

Araújo, C., Henriques, P. R., & Cerqueira, J. J. (2023a). Neuroeducation: Guidelines for computational thinking training. In 2023 International Symposium on Computers in Education (SIIE) (p. 1-6). IEEE Xplore. doi: https://doi.org/10.1109/SIIE59826.2023.10423685

Araújo, C., Henriques, P. R., & Cerqueira, J. J. (2023b). OntoCnE, characterizing Learning Resources for training Computational Thinking. In 2023 International Symposium on Computers in Education (SIIE) (p. 1-6). IEEE Xplore. doi: https://doi.org/10.1109/SIIE59826.2023.10423710

Barefoot. (2021). Computational Thinking Concepts and Approaches. (Accessed: 2021-11-07)

Dalal, M., Carberry, A., & Archambault, L. (2021). Developing a ways of thinking framework for engineering education research. Studies in Engineering Education, 1(2).

Dolgopolovas, V., Dagienė, V., Jasutė, E., & Jevsikova, T. (2019). Design science research for computational thinking in constructionist education: a pragmatist perspective. *Problemos*, 95, 144–159.

García-Peñalvo, F. J. (2018). Editorial Computational Thinking. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, 13(1), 17–19.

Gomes, A., & Mendes, A. J. (2007). Learning to program - difficulties and solutions. In Proceedings of the international conference on engineering education (ICEE-2007) (p. 283-287).

Hemmendinger, D. (2010). A plea for modesty. ACM Inroads, 1(2), 4–7. doi: https://doi.org/10.1145/1805724.1805725

ISTE, & CSTA. (2011). Operational Definition of Computational Thinking for K–12 Education. (Accessed: 2021-11-07)

Lawan, A. A., Abdi, A. S., Abuhassan, A. A., & Khalid, M. S. (2019). What is difficult in learning programming language based on problem-solving skills? In year international conference on advanced science and engineering (ICOASE) (p. 18-22). doi: https://doi.org/10.1109/ICOASE.2019.8723740

Raabe, A. L. A., Brackmann, C. P., & Campos, F. R. (2018). Currículo de referência em tecnologia e computação: da educação infantil ao ensino fundamental (Tech. Rep.). São Paulo: Centro de Inovação para a Educação Brasileira (CIEB). (Accessed: 2023-05-07)

Selby, C., & Woollard, J. (2013). Computational thinking: the developing definition. Retrieved from https://eprints.soton.ac.uk/356481/

Sysło, M. M., & Kwiatkowska, A. B. (2013). Informatics for all high school students. In I. Diethelm & R. T. Mittermeir (Eds.), Informatics in schools. sustainable informatics education for pupils of all ages. ISSEP (Vol. 7780, pp. 43–56). Springer Berlin Heidelberg. doi: https://doi.org/10.1007/978-3-642-36617-8 4

Wing, J. M. (2006). Computational thinking. Commun. ACM, 49(3), 33–35. doi: https://doi.org/10.1145/1118178.1118215

Integrating Eye-Tracking and Artificial Intelligence for Human-Centered Visual Attention Analytics in Online Learning

Alvaro Becerra, Ruth Cobos

Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente, 11, Cantoblanco, 28049 Madrid
{alvaro.becerra,ruth.cobos}@uam.es

Resumen: Este artículo presenta una versión extendida de *Visual Attention Analysis Dashboard* (VAAD), un sistema de analíticas del aprendizaje multimodal que integra datos de seguimiento ocular para analizar la atención visual de los estudiantes en entornos de aprendizaje en línea a través de un *Dasboard*. La nueva versión de VAAD incorpora dos avances principales: (i) un rediseño centrado en el usuario (*Human-Centered Design*) que mejora la interpretabilidad y usabilidad del *Dashboard* tanto para instructores como para científicos de datos, y (ii) modelos predictivos basados en la mirada que generalizan mejor a condiciones reales de aprendizaje. Utilizando el *dataset* IMPROVE, con 120 estudiantes, se desarrollaron y validaron modelos de aprendizaje automático capaces de distinguir entre actividades como ver videos, leer o resolver tareas. Los resultados muestran que las características oculares y pupilares permiten discriminar de manera fiable diferentes comportamientos de aprendizaje, mientras que la nueva interfaz facilita interpretaciones transparentes y pedagógicamente significativas. En conjunto, estas contribuciones refuerzan la relevancia práctica de las analíticas de atención visual y promueven una integración responsable de la IA en la educación.
Palabras clave: Atención visual, analíticas del aprendizaje multimodal, seguimiento ocular, panel de control, diseño centrado en el usuario, inteligencia artificial.

Abstract: This paper presents an extended version of the Visual Attention Analysis Dashboard (VAAD), a multimodal learning analytics system that integrates eye-tracking data to analyse students' visual attention in online learning through a Web-based Dashboard. The new VAAD introduces two main advances: (i) a Human-Centered redesign of the dashboard to improve interpretability and usability for instructors and data scientists, and (ii) enhanced gaze-based predictive models that generalize to real learning conditions. Using the IMPROVE dataset with 120 learners, we developed and validated machine learning models capable of distinguishing between learning activities such as video watching, reading, and assignment solving. Results show that eye- and pupil-based features enable reliable discrimination of learning behaviours, while the redesigned interface facilitates transparent and pedagogically meaningful interpretations. Together, these contributions strengthen the practical relevance of visual attention analytics and advance the responsible integration of AI in education.
Keywords: Visual Attention, Multimodal Learning Analytics, Eye Tracking, Dashboard, Human-Centered Design, Artificial Intelligence.

1. Introduction

Online learning has grown exponentially in recent years, particularly through Massive Open Online Courses (MOOCs), which have made education accessible to a global audience. However, this expansion has also revealed several challenges inherent to large-scale online modalities. Among them, students often report feelings of isolation and a perceived lack of support and feedback from instructors. These issues contribute to a decline in learner motivation, ultimately leading to increased dropout rates (Topali et al., 2019).

Furthermore, recent research has shown that relying exclusively on digital traces captured by MOOC and other online learning platforms (such as clicks, navigation times, or completed exercises) is insufficient to understand how students feel, how much attention they sustain, where they are focusing on, or the cognitive effort they are investing. This has fuelled growing interest in approaches capable of capturing non-visible dimensions of the learning experience, such as visual attention, stress, or cognitive load. In this regard, advances in biosensors and Multimodal Learning Analytics (MMLA) provide new opportunities to monitor students from a more holistic and contextualized perspective, integrating physiological signals with interaction data and advanced computational analysis, as is shown in the literature review by Becerra, Cobos, et al. (2025).

Among the most widely used biosensors, webcams stand out as a primary source of valuable information (Becerra, Cobos, et al., 2025). They have been used, for example, to detect distractions through pose estimation (Becerra, Daza, et al., 2025; Becerra et al., 2024).

The second most commonly used biosensor is the eye tracker (Becerra, Cobos, et al., 2025), which provides detailed information about students' visual attention. From the signals captured by an eye tracker, it is possible to extract fixation patterns (where and for how long learners focus their gaze), saccades (how they move their eyes across the screen), and pupil diameter, a physiological indicator closely related to mental effort. Eye-tracking data have been used, among other applications, to predict learner motivation during online activities (Sharma et al., 2020) and to identify specific areas of instructional content that students find difficult to comprehend (Minematsu et al., 2020). Furthermore, pupil-based measurements have proven valuable for detecting increases in cognitive load and attentional demand, enabling a deeper understanding of how learners engage with learning materials beyond what is visible from platform interaction logs alone (Beatty, 1982; Čegovnik et al., 2018).

However, despite the demonstrated value of eye-tracking data and its potential to provide deeper insights into students' learning processes, there is still a lack of tools that allow educators and researchers to visualize and interpret these signals within their instructional context. Without such contextualized visualizations, raw gaze metrics offer limited utility for understanding how learners interact with online materials or for informing pedagogical decisions. To address this need, we presented VAAD (Visual Attention Analysis Dashboard), a tool designed to provide both descriptive and predictive analytics of visual attention during online learning sessions (Navarro et al., 2024). Building upon our previous work introducing VAAD, this paper presents a substantially extended version of the system. First, we incorporate new visualizations grounded in a Human-Centered Design approach, in which MOOC instructors and data scientists were actively consulted to ensure that the analytical outputs align with real pedagogical and research needs. Second, we enhance the predictive module of VAAD by redesigning its gaze-based activity recognition models to ensure better generalization to real online learning conditions. Through these enhancements, the new version of VAAD not only advances visual attention analytics from a technical standpoint, but also strengthens its relevance and applicability in real online learning environments.

2. Related Works

2.1. Learning Analytics Dashboards

Learning Analytics Dashboards have emerged as a widely adopted approach to support decision-making in online learning environments (Klerkx et al., 2017). These dashboards typically process and display digital interaction data, such as activity logs, assessment results, or participation indicators, to provide actionable insights into learner behaviour for instructors, students, or both. Their primary goal is to make complex learning data intelligible and useful, enabling instructors to monitor progress, detect difficulties, and intervene when necessary, while helping students develop self-regulation skills through timely feedback about their performance and engagement (Cobos, 2023).

Beyond traditional dashboards that rely solely on platform interaction logs, recent advances in MMLA have motivated the development of multimodal dashboards, which integrate behavioural and physiological signals to provide a more comprehensive understanding of the learning process. For example, M2LADS (Becerra et al., 2023) supports the integration and visualization of multimodal data including EEG attention measures, heart rate, and eye-tracking signals in MOOC environments, enabling instructors to analyse learners' experience through synchronized visual feedback and performance indicators.

Regarding eye-tracker–based dashboards, Abeysinghe et al., (2023) proposed a real-time, multi-user eye-tracking visualization system that integrates both traditional gaze metrics and advanced indicators like the ambient/focal attention coefficient K, allowing instructors or proctors to analyse visual search behaviours and collaborative engagement in distributed settings. Similarly, Jayawardana et al (2021) has emphasized the real-time visual presentation of gaze streams through metadata-driven processing pipelines, allowing dynamic rendering of gaze points, heatmaps, and attention-based overlays as data are streamed from heterogeneous eye-tracking devices. In a more recent study (Jayawardena et al., 2024), the same research line has been extended to support the real-time visualization of additional gaze-derived indicators, including cognitive load estimates, gaze transition matrices, and main-sequence relationships, allowing analysts to explore not only where learners look, but also how their attentional state and visual strategy evolve throughout task performance. Additionally, (Andreu-Perez et al., 2016) introduced a dashboard for integrating diverse eye-activity metrics, such as saccades, fixations, pupil dynamics, and blink-related indicators, enhancing the analytical scope.

In this context, it has been increasingly emphasized that the success and adoption of analytic solutions depend not only on their technical sophistication but also on how well they align with the needs, values, and capacities of their intended users. Human-Centered Learning Analytics (HCLA) specifically aims to ensure the active involvement of educational stakeholders throughout the design, development, and evaluation of dashboards and analytic tools, thereby enhancing pedagogical relevance, usability, trustworthiness, and real-world adoption (Topali et al., 2024, 2025).

At the same time, emerging research demonstrates that the integration of generative AI into learning analytics dashboards is enabling richer forms of interpretation and feedback, moving beyond static data visualization toward more interactive, personalized, and explanatory support mechanisms. For example, VizChat augments dashboards with contextualized explanations generated by generative AI, helping users understand visualizations and reduce cognitive overload through conversational guidance (Yan et al., 2024). Similarly, generative models have been incorporated into collaborative feedback systems, such as AICoFe, which couples dashboards with AI-driven qualitative feedback to enhance student reflection and professional skill development in engineering courses (Becerra & Cobos, 2025). In the context of online learning, GePeTo leverages learner interaction data to automatically generate tailored recommendations within dashboard environments, improving support and motivation in large-scale online learning scenarios (Becerra, Mohseni, et al., 2024)

2.2. Eye-Tracking and Visual Attention

Research in cognitive psychology has demonstrated that the human eye provides reliable indicators of attentional and mental processing demands. Beatty (1982) showed that task-evoked pupillary responses systematically increase with processing load, providing a robust physiological measure of "mental effort" during cognitive activities. More recent empirical evidence extends this relationship beyond controlled laboratory tasks: Čegovnik et al. (2018) demonstrated that pupil dilation, blink rate, and fixation patterns captured through an affordable eye-tracking device reliably reflected increasing cognitive load while driving, confirming the sensitivity of ocular signals to changes in attentional demand in dynamic environments. Furthermore, Sankar et al. (2025) demonstrated that fixation-based measures of attention and pupil-based measures of absorption together provide a more holistic representation of how users cognitively engage with complex digital information systems, highlighting that gaze duration and pupil dilation reflect distinct yet interrelated aspects of sustained information processing. Complementing these traditional pupillary measures, Duchowski et al. (2018) introduced the Index of Pupillary Activity (IPA), showing that moment-to-moment oscillations in pupil diameter can discriminate task difficulty and provide a more sensitive estimate of cognitive load than static dilation measures.

2.3. Gaze-Based AI Models

Recent advances in AI-driven gaze analysis have shown that visual attention signals can support the automated prediction of learners' mental and behavioural states. For instance, (Sharma et al., 2020) proposed a set of stimulus-based gaze features to infer motivation and learning gains in online learning contexts, demonstrating that performance can be predicted with less than 5% error by combining heatmap-derived variables with machine learning models. Extending this approach toward affective inference, (Jyotsna et al., 2023) introduced a personalized deep learning model that analyses temporal patterns in eye-tracking signals to estimate mental states such as stress and calmness with high accuracy (86.4%) thanks to modelling of gaze dynamics. Similarly, eye-tracking data have also been used to detect fatigue, a cognitive condition directly related to attentional engagement. (Andreu-Perez et al., 2016) developed several machine learning models that used saccade, fixation, blink and pupil-based measures and demonstrated that combining the most informative features enables distinguishing fatigued vs. rested states with an accuracy of 76.4%.

3. Methodology

3.1. Data Source: IMPROVE Dataset

We used data from the IMPROVE dataset (Daza et al., 2025), which includes recordings from 120 learners during real online HTML learning sessions on an edX MOOC. Although the dataset integrates 16 sensors, this study focuses on two data streams: (i) eye-tracking signals captured at 120 Hz with a Tobii Pro Fusion device, and (ii) learning activity logs (video watching, reading, and assignments). All signals were synchronized with the activity logs to contextualize learners' gaze behaviour within each task.

3.2. Human-Centered Dashboard Design

The redesign of VAAD adopted a Human-Centered Design (HCD) approach to enhance its interpretability, usability, and practical value for visual attention analysis in large-scale online learning. MOOC instructors and data scientists evaluated the previous version of VAAD and provided feedback on the clarity of indicators, interpretation challenges, and missing functionalities. Their input guided iterative prototyping cycles in which new visualizations, graphical improvements, and video-based insights were collaboratively developed to better contextualize learners' gaze behaviour.

3.3. Generalized Gaze-Based Prediction Models

To extend the analytical capabilities of VAAD, we redesigned the predictive module with a strong focus on generalization to real online learning conditions. First, instead of training models on a reduced sample of 80 participants as done in the previous version of VAAD, we leveraged the full set of 120 learners included in the IMPROVE dataset. Second, we expanded the target space by predicting three types of learning activities: video watching, reading, and assignment solving, rather than only distinguishing between video watching and reading. Third, we extracted an extended set of gaze- and pupil-based features. Fourth, we evaluated a wider family of machine learning models than in the original VAAD.

3.2.1. Feature Extraction

Continuous eye-tracking data were divided into non-overlapping time windows corresponding to the student's current learning activity. To examine the effect of temporal context on predictive performance, six window durations (5, 10, 15, 20, 25 and 30 seconds) were tested. Each window was labelled according to the activity: video watching, reading, or solving assignments. For each window, a total of 72 features were extracted, as summarized in Table 1. These biometric descriptors have been widely adopted in machine learning models for physiological signals (Becerra, Daza, et al., 2025; Daza et al., 2024).

Before training the predictive models, we applied a data cleaning and class-balancing procedure. First, windows containing missing values (NaNs) in any feature were removed. Second, to avoid biases caused by an uneven number of windows per activity for each student, the dataset was balanced individually per learner. Specifically, for each student, we identified the minimum number of available windows across the three activity classes (video watching, reading, assignment solving), and randomly downsampled the remaining classes to match that value. For 5-second windows, 9,264 samples were retained (3,088 per activity). For 10-second windows, 4,671 samples remained (1,557 per activity), followed by 3,150 samples for 15-second windows (1,050 per activity), 2,301 samples for 20-second windows (767 per activity), 1,830 samples for 25-second windows (610 per activity), and 1,485 samples for 30-second windows (495 per activity).

3.2.3. Modelling Approach and Validation

We compared four machine learning classifiers widely used in MMLA and optimized their key hyperparameters

- Random Forest: number of trees ∈ {300, 600}, maximum depth ∈ {None, 15}, maximum number of features ∈ {sqrt, log2}.
- Linear SVM: regularization parameter C ∈ {0.1, 1, 10}.
- RBF-SVM: C ∈ {0.5, 2, 8}, kernel scale gamma ∈ {scale, auto}.
- MLP classifier: hidden layer configurations ∈ {(64,), (128,), (64, 32)}, regularization coefficient α ∈ {1e-4, 1e-3}, with ReLU activation and early stopping enabled (max_iter = 300).

Feature dimensionality was handled through multiple feature selection: (1) no feature selection (full feature set), (2) Principal Component Analysis retaining 95% variance, (3) Mutual Information–based selection of the top 20% most informative features, and (4) embedded L1-based selection via sparse logistic regression.

Model evaluation followed a nested cross-validation scheme with participant-wise grouping to avoid data leakage. The outer 5-fold GroupKFold provided unbiased performance estimates, while the inner 3-fold GroupKFold optimized hyperparameters. Given the class-balanced dataset, performance was assessed primarily using the F1-score, complemented by accuracy, precision, and recall.

Category	Feature name
Gaze velocity (33 features)	Total positive velocity, Total negative velocity, Location of first maximum, Location of second maximum, Location of third maximum, Average velocity / |v|max, Average velocity / vmax, RMS of velocity / |v|max, Mean of segment, Standard deviation of segment, RMS acceleration / |a|max, Median of segment, Standard deviation of velocity, Standard deviation of acceleration, Jerk, Average jerk, Max jerk absolute value, Maximum jerk, RMS jerk, Location of maximum absolute jerk, Location of maximum jerk, Number of sign changes, Sum(v > 0)|v| / Sum(v < 0)|v|, Sum(v > 0) / Sum(v < 0), vmax – vmin, Mean normalized, Local maxima count, Average acceleration, Maximum of segment, Minimum of segment, Skewness, Kurtosis
Pupil size (33 features)	Same 33 features computed for pupil diameter signal
Saccade metrics (5 features)	Saccade count, Saccade rate, Mean saccade duration, Saccade time fraction, Mean saccade velocity
Pupillometry cognitive effort (1 feature)	IPA (Index of Pupillary Activity)
Demographic descriptor (1 feature)	Gender

Table 1. Summary of the gaze-, pupillometry-, and saccade-based features extracted per window for activity prediction

4. Human-centered enhancements to VAAD

Based on the requirements gathered from MOOC instructors and data scientists (Section 3.2), we implemented a set of concrete enhancements aimed at (i) improving the interpretability of gaze- and pupil-based indicators, (ii) enabling group-level comparisons of cognitive load, and (iii) providing richer, time-aligned visualizations for individual learner analysis.

(1) Embedded help pop-ups for non-expert users (Figure 3).

One of the most recurrent comments from MOOC instructors was that, although the dashboard presented advanced gaze metrics, some of those metrics (e.g., saccade rate or the Index of Pupillary Activity) were not self-explanatory. To reduce this interpretive gap, we added contextual help pop-ups that appear when hovering or clicking on an indicator. Each pop-up provides: (a) a short definition, (b) the expected range/units, and (c) a short note on how to read the value in the learning context. This enhancement was explicitly requested by instructors to make the dashboard usable without having to consult external documentation.

(2) Extended set of boxplots for cognitive-load comparison (Figure 3).

In the previous version, the global analysis of the dashboard was limited to the number of saccades/fixations and their average duration. Data scientists pointed out that this was not enough to compare cognitive demand across groups, gender, or learning activities. We therefore extended the statistical view with additional boxplots that now include pupil size and IPA. These plots can be faceted by activity type and by demographic groups, so analysts can visually inspect whether certain groups systematically exhibit higher pupil dilation or IPA values.

(3) Reworked individual-learner view with video overlay (Figure 4).

MOOC instructors reported that static heatmaps were of limited usefulness for individual analyses because they did not preserve the temporal evolution of the fixation pattern, nor did they show *what* the student was actually seeing at that moment. To address this, we incorporated a screen-capture video of the session and overlaid a red gaze pointer that moves frame by frame to indicate where the learner was looking. In addition, the heatmap is now computed dynamically over the last X seconds and superimposed on top of the video. This rolling heatmap helps instructors see not only isolated fixations, but how attention gradually moved across relevant areas of the content.

(4) Time-synchronized pupil/IPA traces linked to the video (Figure 4).

To further support joint exploration by instructors and data scientists, we added time-series plots of pupil size and IPA color-coded by the activity the learner was performing at each moment. These plots are fully synchronized with the session video: clicking on a point in the plot moves the video to that timestamp, and playing or scrubbing the video highlights the corresponding datapoint in the plot. This bidirectional synchronization makes it possible to answer questions such as *"during which part of the video did cognitive load peak?"* or *"does the IPA spike coincide with the start of the reading segment?"*.

(5) Visualizations for the prediction module (Figure 5).

The original prediction module only produced performance values, but it did not expose them visually. Following the suggestions of the data scientists, we added several analytical views:

- **Ranking charts** that order models by different metrics for each window size and also in a global view.
- **Boxplots comparing performance across window sizes** to assess the stability of the models when the temporal context changes.
- **Boxplots by feature selector** to see whether dimensionality reduction has a consistent effect across windows or in the global session.
- **ROC curves (Figure 1 and 2)** for the best-performing models per window size and for the global setting, so that model comparison is not limited to a single operating point.

Together, these views make the predictive module inspectable: users can see which models work best under which temporal granularity and with which feature-selection strategy.

5. AI Models Results

Results show a consistent improvement in model performance as the temporal window increases, indicating that longer segments capture more stable behavioral and physiological patterns. In the binary classification task (video watching vs. reading), accuracy rose from 0.71 with 5-second windows to 0.76 with 25-second windows, with the Random Forest model achieving the highest performance at that point. Similarly, in the multiclass task (video watching, reading, and assignment solving), accuracy improved from 0.49 to 0.54, with the RBF-SVM achieving the best results for 25-second windows. In both cases, 25s windows provided the best balance between temporal resolution and signal stability, while performance slightly declined for longer segments (30 s). These results are summarized in Tables 2 and 3. The ROC curves of the best-performing models (Figures 1 and 2) further confirm their strong discriminative ability in both binary and multiclass settings, with consistently high AUC values

Window (s)	Best Model (Features/ Features selector)	Accuracy M (SD)	F1 macro M (SD)
5	svm_rbf (no_gaze / none)	0.709 (0.014)	0.709 (0.014)
10	mlp (all / l1sfm)	0.731 (0.005)	0.731 (0.005)
15	svm_linear (all / l1sfm)	0.733 (0.016)	0.732 (0.015)
20	mlp (all / l1sfm)	0.745 (0.035)	0.745 (0.035)
25	**rf (no_gaze / none)**	**0.764 (0.031)**	**0.763 (0.031)**
30	svm_linear (all / none)	0.757 (0.035)	0.756 (0.036)

Table 2. Summary of the best-performing models and classification results for the binary prediction task across different temporal window sizes

Window (s)	Best Model (Features/ Features selector)	Accuracy M (SD)	F1 macro M (SD)
5	mlp (all / mi20)	0.491 (0.015)	0.481 (0.019)
10	mlp (all / l1sfm)	0.517 (0.007)	0.510 (0.006)
15	svm_linear (all / l1sfm)	0.511 (0.018)	0.502 (0.016)
20	rf (all / none)	0.519 (0.011)	0.513 (0.011)
25	**svm_rbf (all / none)**	**0.539 (0.015)**	**0.536 (0.018)**
30	svm_rbf (no_gaze / l1sfm)	0.523 (0.024)	0.516 (0.030)

Table 3. Summary of the best-performing models and classification results for the multiclass prediction task across different temporal window sizes

6. Discussion

This work extends VAAD in two complementary directions: (i) the redesign of the dashboard through a Human-Centred Design approach, and (ii) the development of gaze-based predictive models that generalize better to realistic online learning conditions. Together, these contributions bridge the gap between technically advanced visual attention analytics and tools that are interpretable and actionable for both instructors and data scientists. The new VAAD supports MOOC instructors with intuitive, pedagogically meaningful visualizations and offers analysts flexible, transparent views for model inspection and refinement.

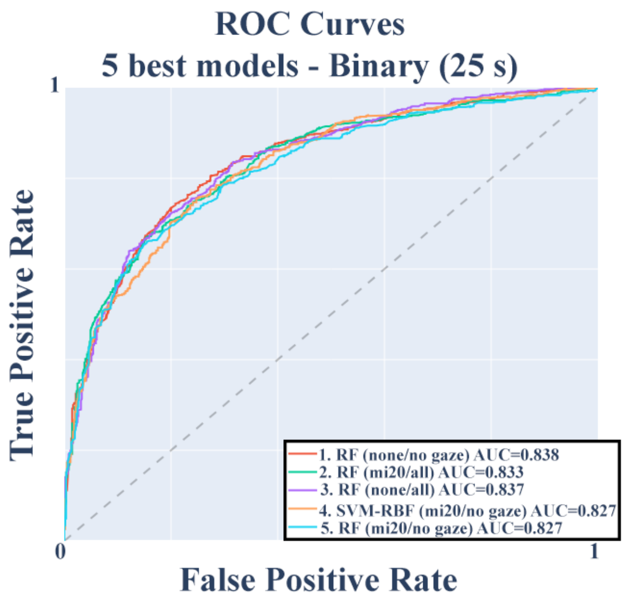


Figure 1 ROC curves of the five best binary classification models (25-second window). The results were extracted from the *Prediction Models* section of the VAAD Dashboard

From a design perspective, the introduced enhancements directly address adoption barriers reported in Human-Centred Learning Analytics literature. Embedded help pop-ups, extended boxplots for cognitive load, and a redesigned individual-learner view with video overlay make complex gaze and pupil metrics interpretable to non-expert users. For instructors, these features translate technical indicators such as saccade rate or IPA into meaningful signals of attention and cognitive demand, while for data scientists they facilitate exploration of group-level patterns and validation of modelling assumptions.

The new predictive visualizations also enhance transparency and trust in AI-based analytics. Ranking charts, boxplots, and ROC curves allow users to compare model performance across window sizes and feature-selection strategies, clarifying how design choices affect outcomes. From a modelling standpoint, results confirm that eye-tracking and pupillometry signals provide meaningful information to distinguish learning activities in online settings. Performance improved steadily with longer temporal windows, reaching up to 0.76 accuracy in the binary (video watching vs. reading) task and around 0.54 in the multiclass case (video watching, reading, and assignment solving). These findings suggest that short windows are too sensitive to transient fluctuations, while moderate windows (20–25 s) capture more stable patterns of attention and cognitive effort.

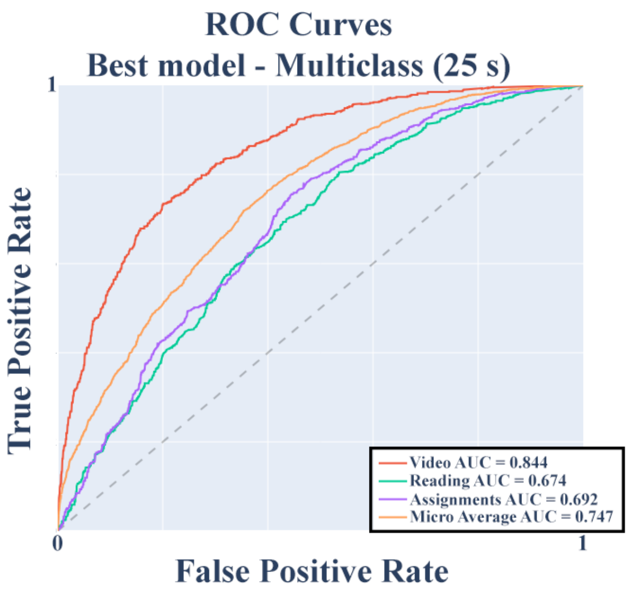


Figure 2 ROC curves of the best multiclass classification model (25-second window). The results were extracted from the *Prediction Models* section of the VAAD Dashboard

The ROC curves (Figures 1 and 2) provide additional insights into the discriminative capacity of the models beyond overall accuracy and F1-scores. In the binary classification task, the curves show consistently high separability across the top five models, with areas under the curve (AUC) above 0.85 for all cases. This confirms that the models are robust in distinguishing between both activities. In the multiclass scenario, the ROC analysis reveals a clear hierarchy of predictability among activities. The *video-watching* class exhibits the highest AUC, indicating that its characteristic gaze and pupillary patterns are more distinct. In contrast, *reading* and *assignment-solving* show more overlapping response distributions, which explains their lower AUC values and the moderate confusion observed between these two classes. The micro-average ROC consolidates these findings, suggesting that overall, the model maintains a balanced trade-off between sensitivity and specificity across all activity types.

Comparing algorithms and feature configurations reveals that MLP models perform better on short, noisy sequences, while SVMs and Random Forests excel with longer windows. Interestingly, some of the best-performing binary models did not rely on gaze-position features, whereas multiclass models achieved their highest performance when using the full feature set. This suggests that, for coarse distinctions between two activities, global velocity and pupillary dynamics may be sufficient, while more fine-grained activity discrimination benefits from richer combinations of gaze, saccade, and pupil-based descriptors.

The predictive experiments presented in this work serve as a proof of concept for using gaze-based models to automatically label learning activities without continuous human annotation. In large-scale eye-tracking datasets, where manual labelling is impractical, such models can provide initial activity labels that are later refined or audited, significantly reducing annotation effort. Importantly, VAAD's prediction module is designed to be model-agnostic, allowing the visualization of outputs from any compatible classifier or regressor. This flexibility not only supports the integration of new algorithms, feature sets, or target variables but also enhances the scalability and longevity of the system.

7. Conclusions and Future Work

This study presented an extended version of the Visual Attention Analysis Dashboard (VAAD) (Navarro et al., 2024), integrating a Human-Centered redesign with gaze-based predictive models capable of generalizing to real online learning contexts. The results demonstrate that eye-tracking and pupillometry data contain rich behavioral information that can be effectively leveraged to infer learners' activities and visual attention patterns. Beyond technical improvements, the new VAAD bridges the gap between advanced multimodal analytics and pedagogically meaningful tools, making visual attention data interpretable, actionable, and transparent for both instructors and data scientists. By combining explainable visualizations with machine learning models, VAAD contributes to the development of responsible and human-centered AI for education.

From a broader perspective, this work reinforces the potential of multimodal learning analytics to move beyond log-based indicators and capture the hidden dimensions of learners' cognitive and affective states. In doing so, VAAD supports a more holistic understanding of engagement, which is crucial for addressing persistent challenges in online and MOOC-based education such as disengagement, low motivation, and dropout. The integration of gaze-based indicators with interpretable visual dashboards represents a concrete step toward making complex AI systems accessible to educators and aligning technological innovation with real pedagogical needs.

Nevertheless, several aspects remain open for future work. First, further validation of the predictive models with more diverse learner populations and learning platforms is needed to confirm the generalizability of the results. Second, new research should explore how instructors actually interpret and use VAAD's feedback in their teaching practice, evaluating its impact on instructional design and learner outcomes. Third, we envision extending VAAD to in-person monitoring scenarios using eye-tracking glasses (Becerra, Andres, et al., 2025), enabling the continuous observation of students' visual attention in physical classrooms and hybrid learning environments.

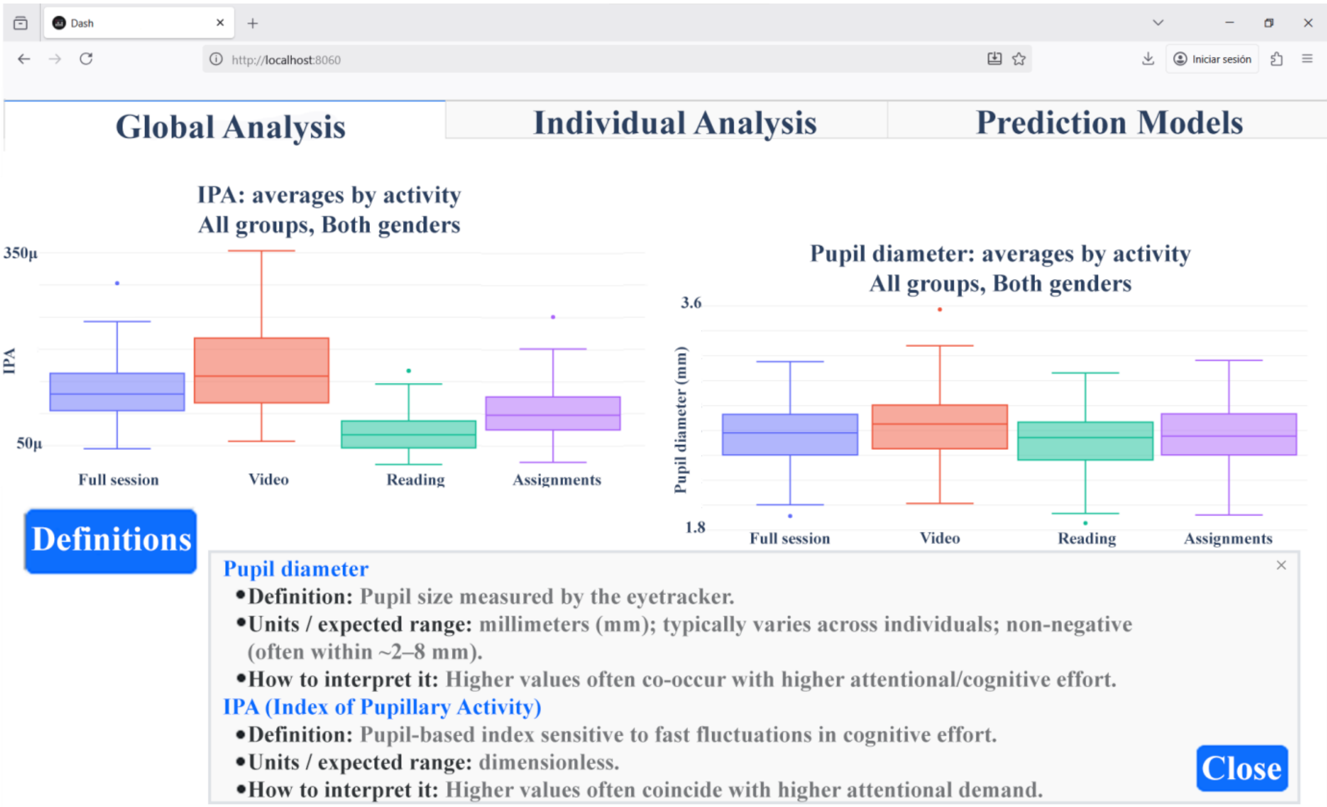


Figure 3. Global Analysis section from the VAAD Dashboard

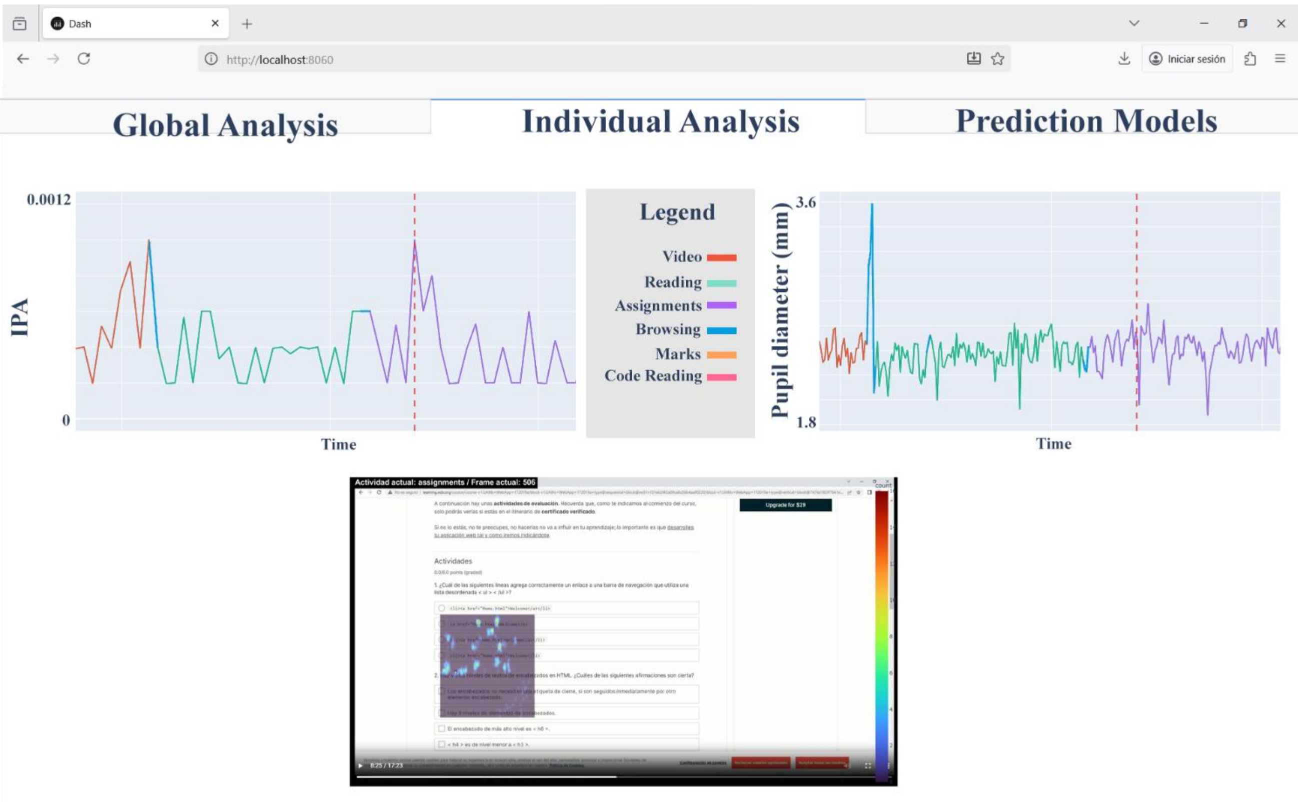


Figure 4. Individual Analysis section from the VAAD Dashboard

Figure 5. Prediction Models section from the VAAD Dashboard

Acknowledgements

Support by projects: Cátedra ENIA UAM-VERIDAS en IA Responsable (NextGenerationEU PRTR TSI-100927-2023-2), M2RAI (PID2024-160053OB-I00, MICIU/FEDER) and SNOLA (RED2022-134284-T).

The authors also thank Miriam Navarro for her contribution to the initial development of VAAD

References

Abeysinghe, Y., Mahanama, B., Jayawardena, G., Sunkara, M., Ashok, V., & Jayarathna, S. (2023). Gaze Analytics Dashboard for Distributed Eye Tracking. *2023 IEEE 24th International Conference on Information Reuse and Integration for Data Science (IRI)*, 140-145. https://doi.org/10.1109/IRI58017.2023.00031

Andreu-Perez, J., Solnais, C., & Sriskandarajah, K. (2016). EALab (Eye Activity Lab): A MATLAB Toolbox for Variable Extraction, Multivariate Analysis and Classification of Eye-Movement Data. *Neuroinformatics*, *14*(1), 51-67. https://doi.org/10.1007/s12021-015-9275-4

Beatty, J. (1982). Task-evoked pupillary responses, processing load, and the structure of processing resources. *Psychological bulletin*, *91*(2), 276.

Becerra, A., Andres, D., Villegas, P., Daza, R., & Cobos, R. (2025). *MOSAIC-F: A Framework for Enhancing Students' Oral Presentation Skills through Personalized Feedback*. https://arxiv.org/abs/2506.08634

Becerra, A., & Cobos, R. (2025). Enhancing the Professional Development of Engineering Students through an AI-Based Collaborative Feedback System. *2025 IEEE Global Engineering Education Conference (EDUCON)*, 1-9. https://doi.org/10.1109/EDUCON62633.2025.11016499

Becerra, A., Cobos, R., & Lang, C. (2025). Enhancing online learning by integrating biosensors and multimodal learning analytics for detecting and predicting student behaviour: A review. *Behaviour & Information Technology*, *0*(0), 1-26. https://doi.org/10.1080/0144929X.2025.2562322

Becerra, A., Daza, R., Cobos, R., Morales, A., Cukurova, M., & Fierrez, J. (2023). M2lads: A system for generating multimodal learning analytics dashboards. *2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC)*, 1564-1569. https://doi.org/10.1109/COMPSAC57700.2023.00241

Becerra, A., Daza, R., Cobos, R., Morales, A., Cukurova, M., & Fierrez, J. (2025). AI-Based Multimodal Biometrics for Detecting Smartphone Distractions: Application to Online Learning. En K. Tammets, S. Sosnovsky, R. Ferreira Mello, G. Pishtari, & T. Nazaretsky (Eds.), *Two Decades of TEL. From Lessons Learnt to Challenges Ahead* (pp. 31-46). Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-03870-8_3

Becerra, A., Irigoyen, J., Daza, R., Cobos, R., Morales, A., Fierrez, J., & Cukurova, M. (2024). Biometrics and Behavior Analysis for Detecting Distractions in e-Learning. *2024 International Symposium on Computers in Education (SIIE)*, 1-6. https://doi.org/10.1109/SIIE63180.2024.10604582

Becerra, A., Mohseni, Z., Sanz, J., & Cobos, R. (2024). A generative ai-based personalized guidance tool for enhancing the feedback to mooc learners. *2024 IEEE Global Engineering Education Conference (EDUCON)*, 1-8. https://doi.org/10.1109/EDUCON60312.2024.10578809

Čegovnik, T., Stojmenova, K., Jakus, G., & Sodnik, J. (2018). An analysis of the suitability of a low-cost eye tracker for assessing the cognitive load of drivers. *Applied ergonomics*, *68*, 1-11.

Cobos, R. (2023). Self-Regulated Learning and Active Feedback of MOOC Learners Supported by the Intervention Strategy of a Learning Analytics System. *Electronics*, *12*(15), 3368. https://doi.org/10.3390/electronics12153368

Daza, R., Becerra, A., Cobos, R., Fierrez, J., & Morales, A. (2025). A multimodal dataset for understanding the impact of mobile phones on remote online virtual education. *Scientific Data*, *12*(1), 1332. https://doi.org/10.1038/s41597-025-05681-7

Daza, R., Gomez, L. F., Fierrez, J., Morales, A., Tolosana, R., & Ortega-Garcia, J. (2024).

DeepFace-Attention: Enhancing Attention Level Estimation in E-Learning Environments using Face Biometrics. *IEEE Access*. https://doi.org/10.1109/ACCESS.2024.3437291

Duchowski, A. T., Krejtz, K., Krejtz, I., Biele, C., Niedzielska, A., Kiefer, P., Raubal, M., & Giannopoulos, I. (2018). The Index of Pupillary Activity: Measuring Cognitive Load vis-à-vis Task Difficulty with Pupil Oscillation. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1-13. https://doi.org/10.1145/3173574.3173856

Jayawardana, Y., Jayawardena, G., Duchowski, A. T., & Jayarathna, S. (2021). Metadata-driven eye tracking for real-time applications. *Proceedings of the 21st ACM Symposium on Document Engineering*. https://doi.org/10.1145/3469096.3474935

Jayawardena, G., Ashok, V., & Jayarathna, S. (2024). Advanced Gaze Analytics Dashboard. *2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI)*, 114-119. https://doi.org/10.1109/IRI62200.2024.00034

Jyotsna, C., Amudha, J., Ram, A., Fruet, D., & Nollo, G. (2023). PredictEYE: Personalized Time Series Model for Mental State Prediction Using Eye Tracking. *IEEE Access*, *11*, 128383-128409. https://doi.org/10.1109/ACCESS.2023.3332762

Klerkx, J., Verbert, K., & Duval, E. (2017). *Learning analytics dashboards*.

Navarro, M., Becerra, A., Daza, R., Cobos, R., Morales, A., & Fierrez, J. (2024). VAAD: Visual Attention Analysis Dashboard Applied to e-Learning. *2024 International Symposium on Computers in Education (SIIE)*, 1-6. https://doi.org/10.1109/SIIE63180.2024.10604520

Sankar, G., Djamasbi, S., Tulu, B., & Muehlschlegel, S. (2025). Measuring Cognitive Engagement with Eye-Tracking: An Exploratory Study. *International Conference on Human-Computer Interaction*, 69-78.

Sharma, K., Giannakos, M., & Dillenbourg, P. (2020). Eye-tracking and artificial intelligence to enhance motivation and learning. *Smart Learning Environments*, *7*, 1-19. https://doi.org/10.1186/s40561-020-00122-x

Topali, P., Cobos, R., Agirre-Uribarren, U., Martínez-Monés, A., & Villagrá-Sobrino, S. (2024). 'Instructor in action': Co-design and evaluation of human-centred LA -informed feedback in MOOCs. *Journal of Computer Assisted Learning*, *40*(6), 3149-3166. https://doi.org/10.1111/jcal.13057

Topali, P., Ortega-Arranz, A., Er, E., Martínez-Monés, A., Villagrá-Sobrino, S. L., & Dimitriadis, Y. (2019). Exploring the problems experienced by learners in a MOOC implementing active learning pedagogies. *International Conference on Learning and Collaboration Technologies*, *11475*, 81-90. https://doi.org/10.1007/978-3-030-19875-6_10

Topali, P., Ortega-Arranz, A., Rodríguez-Triana, M. J., Er, E., Khalil, M., & Akçapınar, G. (2025). Designing human-centered learning analytics and artificial intelligence in education solutions: A systematic literature review. *Behaviour & Information Technology*, *44*(5), 1071-1098. https://doi.org/10.1080/0144929X.2024.2345295

Yan, L., Zhao, L., Echeverria, V., Jin, Y., Alfredo, R., Li, X., Gaševi'c, D., & Martinez-Maldonado, R. (2024). VizChat: Enhancing learning analytics dashboards with contextualised explanations using multimodal generative AI chatbots. *International Conference on Artificial Intelligence in Education*, 180-193.