

Estás en la EI1039

Carlos Granell

Sep 16, 2024

Table of contents

Objetivo y resultados de aprendizaje	1
Método de instrucción	1
Materiales y contenido	3
Métodos de evaluación	4
Bibliography	4

Objetivo y resultados de aprendizaje

Según el SIA,

“la asignatura Diseño de Software se imparte en el primer semestre de cuarto curso del grado en Ingeniería Informática de la Universitat Jaume I, como parte del itinerario de Ingeniería del Software. El objetivo de esta asignatura es proporcionar al alumnado conocimientos y habilidades sobre las distintas etapas del diseño del software dentro del proceso de desarrollo.”

Y al finalizar satisfactoriamente la EI1039, deberías ser capaz de:

- Comparar las distintas estrategias que se pueden utilizar en el diseño de un sistema software.
- Seleccionar, aplicar y evaluar patrones de arquitectura y diseño adecuados a la resolución de un determinado problema de desarrollo del software.
- Diseñar y evaluar la interfaz de usuario utilizando principios y técnicas propias de la interacción persona-computador.

Tan importantes, o incluso más, que los resultados de aprendizaje mencionados anteriormente son las competencias blandas o habilidades interpersonales (*soft skills*), como la comunicación, el trabajo en equipo, la síntesis de información, la toma de decisiones, la resolución de conflictos, la gestión del tiempo, y el liderazgo. Trabajaremos en estas competencias o habilidades a lo largo del curso porque, aunque parezca increíble, son aspectos vitales del día a día de un arquitecto/a o diseñador/a de software.

Arquitectura software, patrones de diseño, comunicación, trabajo en equipo, Suena bien, ¿no? Espero que encuentres atractivo e interesante el contenido de la asignatura a lo largo del semestre.

Método de instrucción

💡 ¿Te has preguntado alguna vez qué pasa en tu cerebro cuando aprendes? [1]

Tenemos millones de neuronas que “hacen amigos fácilmente”. Las neuronas se conectan con otras neuronas con cada nueva experiencia que tenemos (nuevas emociones, pensamientos, acciones, conocimiento, etc.) formando redes. Las redes al principio son débiles, pero cada vez que repetimos la experiencia o acción, el camino de la red se consolida y se hace más fuerte. **Cuando aprendemos, nuestro cerebro cambia** por la formación de nuevas conexiones entre neuronas.

¿Quieres saber más? 10 Things Software Developers Should Learn about Learning [2]

💡 Pirámide Bloom

Cuando **reflexionamos** y **aplicamos** (en contraposición a simplemente leer o subrayar, ver Figure 1) cierto conocimiento a contextos nuevos, lo comprendemos más profundamente, provocando que la red sea más densa y tenga conexiones con otras redes. La diferencia entre la red de un principiante y la de un experto radica justamente en el número y densidad de conexiones. **Cuanto más interconexiones, mayor comprensión.**

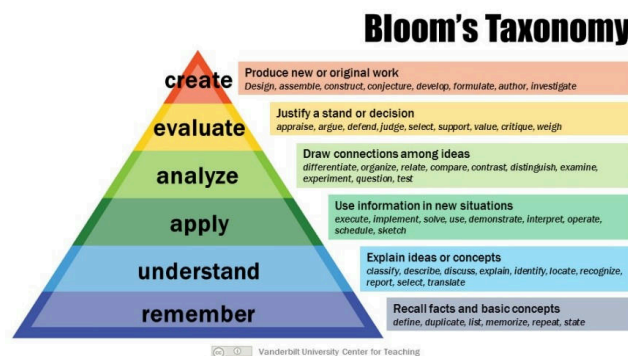


Figure 1: Taxonomía de Bloom

Si esperas presentaciones largas y densas por mi parte todos los viernes por la tarde, entonces creo que te has matriculado de la asignatura equivocada. 😊

Hay evidencia científica de sobra que demuestra una y otra vez que los métodos de *aprendizaje activo* (por *activo* me refiero a todos, tanto dentro como fuera de clase) son mucho más efectivos que escucharme y tomar apuntes de forma pasiva. Vale, es cierto que a veces son necesarias presentaciones aclaratorias; pero tú debes ser el protagonista (y no yo) de tu propio proceso de aprendizaje para alcanzar los resultados esperados con éxito. Por lo tanto, esta asignatura mezcla diversas estrategias de aprendizaje, algunas más tradicionales como presentaciones cuando sea necesario combinadas con estrategias de aprendizaje colaborativas y activas para trabajo en grupo, exploración y aplicación de conceptos, y competencias de proceso (o process skills), como por ejemplo *Flipped Classroom* (lecturas y ejercicios básicos fuera del aula que requieren competencias cognitivas bajas, con análisis y resolución de problemas en aula que requieren competencias cognitivas altas).

Para que te hagas una idea, una semana típica de clase de teoría podría ser así:

- Hasta el miércoles (a las 9:00): Fecha límite para entregar las actividades previas a la siguiente clase de teoría de forma individual en el Aula Virtual.
- Viernes (en aula): Comentamos las actividades previas, junto con presentaciones y actividades en clase.

Las asignaturas EI1039 (Diseño de software) y EI1048 (Paradigmas de Software) están muy relacionadas, ya que son dos caras de la misma moneda a la hora de diseñar y desarrollar aplicaciones avanzadas. Para facilitar el aprendizaje de las competencias de ambas asignaturas, el profesorado nos hemos organizado y coordinado para proponerte un proyecto común que viene detallado en un documento separado colgado en el Aula Virtual de las dos asignaturas. Las clases de laboratorio (LA) de la EI1039 y de la EI1048 son de trabajo (en grupo) para el desarrollo del proyecto común.

Llevamos varios años colaborando entre el profesorado de ambas asignaturas y presentando los resultados en foros sobre educación universitaria en informática (JENUI):

- [3],
- [4],
- [5] y
- [6].

En la edición de la JENUI 2024, se reconoció nuestro trabajo en [5] con el premio SISTEDES al mejor trabajo sobre ingeniería del software.

Materiales y contenido

Los materiales del curso relativos a Teoría (TE) están disponibles en el Aula Virtual, organizados por semanas. Obviamente, el AV enlaza con este sitio web de la asignatura. Desde el punto de vista del estudiantes, es conveniente tener tanto el AV como este sitio web abiertos.

En el Aula Virtual de la asignatura se especifican las actividades previas a título individual que se entregan a través de tareas correspondientes. Voy a utilizar el Aula Virtual para proporcionarte feedback sobre las actividades propuestas, para anuncios de la asignatura y, en definitiva, para cualquier tipo de comunicación oficial. Agradecería enormemente que todas las comunicaciones entre nosotros relativas a la asignatura fueran canalizadas a través del Aula Virtual, y **no a través de mi correo personal**.

La asignatura también consta de sesiones de seminarios (SE) donde exploraremos temas adicionales al diseño de software, como arquitectura de software o comunicación oral, relacionados de algún modo con el proyecto común a desarrollar. La información de los seminarios se encuentra únicamente en el Aula Virtual. Las cinco sesiones de seminarios se organizan como sigue:

- **Seminario 1** (26 septiembre) - MVC y variantes + LLM
- **Seminario 2** (17 octubre) - Caso de estudio: Software architecture patterns
- **Seminario 3** (30/31 octubre) - Conjunto con EI1048. Relacionado con el proyecto común
- **Seminario 4** (20/21 noviembre) - Conjunto con EI1048. Relacionado con el proyecto común
- **Seminario 5** (12 diciembre) - Comunicación oral efectiva

Métodos de evaluación

Vuestra participación en clase es fundamental. Las actividades propuestas en el aula durante las sesiones de teoría invitan al trabajo colaborativo y participativo, fomentando el aprendizaje activo, la discusión y la comunicación. Las actividades previas cuentan un 10%. Es necesario entregarlas todas. Los seminarios computan otro 20% de la nota final, según se desglosa en Table 1. A lo largo del curso se informará con más detalle de la naturaleza de los seminarios evaluables.

El desarrollo, entrega y defensa del proyecto común cubre el 70% de la nota.

Table 1: Instrumentos de evaluación

Actividad		Peso	Tipo	Instrumento
Actividades previas	Individual	10%	Formativa	Código y/o reflexión personal
Seminarios 1 y 2	Individual	10%	Formativa	Reflexión personal
Seminarios 3 y 4 (con EI1048)	Grupo	10%	Formativa	Presentación y discusión
Proyecto común (con EI1048)	Grupo	70%	Acreditativa	Código, documento escrito, presentación y discusión

Política de entrega tardía/retrasada: Las fechas límite semanales (miércoles mediodía para las actividades previas a clase) tienen sentido para que pueda evaluar el trabajo y proporcionar feedback rápido en la siguiente clase de teoría (viernes). Por lo tanto, las actividades entregadas con retraso no se aceptarán sin permiso especial o debida justificación.

Bibliography

- [1] J. M. Lang, *Small Teaching*. Jossey-Bass, 2016. [Online]. Available: <https://www.jamesmlang.com/books>
- [2] N. C. C. Brown, F. F. J. Hermans, and L. E. Margulieux, “10 Things Software Developers Should Learn about Learning,” *Communications of the ACM*, vol. 67, no. 1, pp. 78–87, Dec. 2023, doi: 10.1145/3584859.
- [3] A. González-Pérez, R. A. M. Cárdenas, and D. Llorens Piñana, “Aprendizaje basado en metodologías ágiles centradas en diseño evolutivo dirigido por pruebas de aceptación,” in *Actas de las XXVII Jornadas sobre la Enseñanza universitaria de la Informática (JENUI)*, AENUI, 2021, pp. 99–106. [Online]. Available: https://aenui.org/actas/pdf/JENUI_2021_012.pdf
- [4] A. González-Pérez, C. Granell-Canut, and R. A. M. Cárdenas, “Coordinación de asignaturas dirigida por un proyecto de desarrollo ágil con evaluación unificada,” in *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI)*, AENUI, Jul. 2022, pp. 127–134. [Online]. Available: https://aenui.org/actas/pdf/JENUI_2022_017.pdf

- [5] M. Matey-Sanz, C. Granell-Canut, and R. A. M. Cárdenas, “Estrategias de control y seguimiento activo de proyectos de desarrollo de software,” in *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI)*, AENUI, Jul. 2024, pp. 165–172. [Online]. Available: https://aenui.org/actas/pdf/JENUI_2024_021.pdf
- [6] M. Matey-Sanz, C. Granell-Canut, and R. A. M. Cárdenas, “La IA generativa como acompañante en el ciclo de vida del software,” in *Actas de las Jornadas sobre Enseñanza Universitaria de la Informática (JENUI)*, AENUI, Jul. 2025, pp. 45–53. [Online]. Available: https://aenui.org/actas/fichas/JENUI_2025_005.html