

## ISWD523 DISEÑO DE SOFTWARE GR2SW

**Curso:** GR2SW  
**Fecha:** 11/06/2025  
**Tema:** Planteamiento del proyecto

**Presentación:**

[https://www.canva.com/design/DAGqAfePhDY/VBastF\\_Hz4jv4fOWb7oQJQ/edit?utm\\_content=DAGqAfePhDY&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGqAfePhDY/VBastF_Hz4jv4fOWb7oQJQ/edit?utm_content=DAGqAfePhDY&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)

---

**Contenido:**

Definición Grupo de Estudio .....	2
Encontrar Problemática .....	2
Definición Participantes .....	5
Definición Necesidades .....	7
Requerimientos .....	13
Metodología.....	15
Casos de Uso.....	18
Módulos .....	29
Referencias .....	33
Anexos: .....	33

## Definición Grupo de Estudio

El grupo de estudio está compuesto por estudiantes de ingeniería de sistemas en la Escuela Politécnica Nacional del Ecuador, quienes actualmente cursan un programa de aprendizaje como parte de su formación académica. Los participantes fueron encuestados en línea mediante un formulario que recopiló datos cuantitativos y cualitativos sobre sus experiencias, desafíos y percepciones durante el aprendizaje del programa.

Las encuestas revelan una variedad de niveles de experiencia, desde principiantes hasta intermedios, lo que refleja una muestra representativa de estudiantes en diversas etapas de su educación.

## Encontrar Problemática

Según Sommerville “Las encuestas son útiles para recolectar datos de una población grande, especialmente cuando se requiere conocer actitudes, preferencias o experiencias pasadas de los usuarios. Son un medio económico y eficaz para reunir información que puede informar decisiones de diseño.” [1] Y por eso se elaboró una encuesta para poder identificar las dificultades en el aprendizaje de programación, y a través del análisis de las respuestas de las preguntas de la encuesta (Anexo 1) se obtuvieron las siguientes problemáticas:

### 1. Experiencia de aprendizaje

El 54.5% de los estudiantes califican su experiencia actual aprendiendo a programar en la universidad como "Regular", seguido de un 22.7% que la considera "Insuficiente" y un 4.5% como "Frustrante". Esto indica una insatisfacción generalizada con el proceso de enseñanza-aprendizaje de la programación en la universidad. Muy pocos estudiantes perciben su experiencia como positiva, lo que sugiere una necesidad urgente de revisar y mejorar las metodologías empleadas.

### 2. Dificultades en el aprendizaje

Los aspectos más difíciles del aprendizaje de la programación son:

- *Recordar la sintaxis* (59.1%)
- *Comprender la lógica de programación* (50%)
- *Resolver ejercicios por cuenta propia y Falta de práctica* (ambos con 40.9%)
- Además, *el entendimiento de la orientación a objetos* (36.4%) y *la falta de guía o acompañamiento* (31.8%) también destacan.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

El dominio técnico básico y la autonomía en la resolución de problemas son las principales barreras. Esto evidencia que los estudiantes no solo enfrentan desafíos conceptuales, sino también una carencia de práctica guiada y acompañamiento pedagógico constante.

### 3. Situaciones Experimentadas

Las principales dificultades que los estudiantes han experimentado al aprender a programar son:

- Sentir que avanzan muy lento respecto a los demás (54.5%)
- No entender lo que explica el profesor y confundirse con los ejemplos (ambos 40.9%)
- No saber por dónde empezar a estudiar (36.4%)
- Frustración por no saber si el código está bien (31.8%)
- Miedo a participar por temor a equivocarse (27.3%)

Existe una combinación de inseguridad académica y emocional. Muchos estudiantes sienten que no progresan al mismo ritmo que sus compañeros y tienen dificultades para seguir las explicaciones del docente, lo que genera frustración, dudas y temor a participar.

### 4. Suficiencia del contenido de las clases

El 59.1% de los estudiantes considera que el contenido de las clases *no es suficiente* para aprender a programar de forma efectiva. Además, un 27.3% opina que el contenido solo es *parcialmente adecuado*.

Casi 9 de cada 10 estudiantes sienten que el material impartido en clase no cubre adecuadamente sus necesidades de aprendizaje. Esto sugiere que el programa académico actual requiere una revisión y posiblemente un rediseño para hacerlo más completo y efectivo.

### 5. Recursos Externos

Los estudiantes dependen en gran medida de recursos no institucionales para aprender programación:

- a. ChatGPT u otra IA es el recurso más utilizado (86.4%)
- b. seguido por Videos de YouTube (68.2%)
- c. Cursos externos como Udemy o Platzi también son mencionados por un 36.4%
- d. En contraste, pocos usan Libros o PDFs (9.1%) o piden ayuda a amigos o compañeros (27.3%)

Existe una fuerte inclinación hacia recursos interactivos y automatizados, lo que podría deberse a su disponibilidad, inmediatez o facilidad de acceso. Esto indica que los materiales y métodos ofrecidos en el aula no están satisfaciendo plenamente las necesidades de los estudiantes.

### 6. Impedimento para practicar

## ISWD523 DISEÑO DE SOFTWARE GR2SW

Las principales razones que impiden a los estudiantes practicar programación fuera del aula son:

- a. Falta de tiempo (50%)
- b. No entender por dónde empezar (45.5%)
- c. Falta de motivación (40.9%)
- d. También destacan la ausencia de buenos ejercicios y aburrimiento (ambos con 31.8%)

Los estudiantes enfrentan tanto barreras prácticas como emocionales para continuar aprendiendo por su cuenta. Esto refleja la necesidad de orientación inicial clara, actividades más atractivas y una mejor gestión del tiempo para fomentar la práctica autónoma.

### 7. Identificación de errores

El 50% de los estudiantes indica que *a veces* les resulta difícil identificar por qué sus programas no funcionan, y un 31.8% afirma que *siempre* les resulta difícil. Ningún estudiante respondió que comprende bien sus errores.

Existe una clara debilidad en las habilidades de depuración y análisis de errores. La ausencia de estudiantes que comprendan completamente sus fallos sugiere la necesidad de reforzar estrategias para el razonamiento lógico, seguimiento de ejecución y uso de herramientas de debugging.

### 8. Recursos en línea

El 68.2% de los estudiantes considera que *algunos* recursos en línea son demasiado avanzados o están mal explicados, mientras que un 13.6% opina que *la mayoría* lo están.

Aunque los estudiantes recurren ampliamente a recursos en línea, una parte significativa siente que estos no están bien adaptados a su nivel de conocimiento. Esto indica una falta de materiales educativos accesibles y claros, lo que puede generar frustración o abandono del aprendizaje autónomo.

### 9. Formatos Útiles

Los formatos que los estudiantes consideran más útiles para entender la programación son:

- *Ejercicios interactivos* (63.6%)
- *Videos* (54.5%)
- En contraste, formatos como *lecturas escritas* (18.2%) y *clases presenciales* (31.8%) son menos valorados.

Los estudiantes prefieren enfoques dinámicos y visuales sobre métodos tradicionales. Esto indica la necesidad de incorporar más recursos interactivos y audiovisuales en el proceso de enseñanza para aumentar la efectividad del aprendizaje.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

### 10. Desmotivación

El 45.5% de los estudiantes ha sentido *muchas veces* desmotivación para seguir aprendiendo a programar. Además, un 36.4% indicó que se ha sentido así *a veces*.

Más del 80% de los estudiantes ha experimentado desmotivación en algún grado. Esto refleja un problema emocional relevante que puede estar relacionado con dificultades académicas, falta de apoyo, frustración continua o una percepción de bajo progreso. Requiere estrategias que refuercen la motivación, el acompañamiento y el reconocimiento del avance individual.

## Definición Participantes

Para la definición de participantes, se va a tomar del libro de Analytics Techniques, 2º Edición, por James Cadle, Debra Paul and Paul Turner.

### Identificación de Participantes.

La identificación de las partes interesadas implica considerar a todos los grupos principales que podrían tener interés en el proyecto a realizar. Esto se realiza para identificar a las partes interesadas dentro de estos grupos que puedan tener vínculos o intereses laborales con el área investigada. (pag 79, Cadle, Paul, Turner)

**Recursos:** Encuesta.

**Técnica usada para identificarlos:** Rueda de las partes interesadas (StakeHolders Wheel)

La principal técnica que recomendamos para facilitar la identificación sistemática de las partes interesadas es la rueda de las partes interesadas.

Esta rueda identifica la gama de grupos de partes interesadas y estructura el proceso de identificación. (pag 82)

Participante identificado	Grupo de la Wheel que le corresponde	Justificación
Estudiantes de Ingeniería en Sistemas	Customers / End users	Son los usuarios finales del sistema que se desarrolla. Se beneficiarán directamente de

## ISWD523 DISEÑO DE SOFTWARE GR2SW

		la solución para mejorar su aprendizaje de programación.
Estudiantes de otras carreras	<b>End users</b>	También son usuarios finales. Aunque no del núcleo principal, usarán el sistema, por tanto son clientes del sistema.
Equipo desarrollador	<b>Developers / Partners</b>	Son quienes crean el sistema. Técnicamente podrían verse como “empleados” en la analogía organizacional, o como “partners” si se ve como colaboración.
Ing. Pamela Flores (docente y supervisora)	<b>Managers / Regulators</b>	Aunque no gestiona como una gerente real, <b>supervisa, evalúa el cumplimiento</b> de objetivos y <b>regula</b> lo que se acepta o no en el proyecto.
Docentes de programación	<b>Partners u Other stakeholders</b>	No usarán el sistema directamente, pero pueden <b>colaborar</b> en su uso y promoción, o beneficiarse como parte del ecosistema educativo.

Cuadrícula de poder/interés para clasificación de los StakeHolders:

La cuadrícula de poder/interés es una matriz bidimensional en la que se representan las partes interesadas. (pag 85)

Participante	Grupo (Stakeholder Wheel)	Tipo de Participante	Nivel de Poder	Nivel de Interés	Rol clave en el proyecto
Estudiantes de Ingeniería en Sistemas	Customers / End users	Directo	Bajo	Alto	Usuarios finales del sistema
Estudiantes de otras carreras	End users	Directo	Bajo	Medio/ Alto	Usuarios complementarios
Curso desarrollador (equipo)	Developers / Partners	Directo	Medio	Alto	Desarrollo del sistema
Ing. Pamela Flores	Managers / Regulators	Indirecto	Alto	Alto	Supervisión, evaluación y aprobación académica

## ISWD523 DISEÑO DE SOFTWARE GR2SW

Docentes de programación	Partners u other StakeHolders	Indirecto	Medio	Medio	Uso complementario, retroalimentación
--------------------------	-------------------------------	-----------	-------	-------	---------------------------------------

## Definición Necesidades

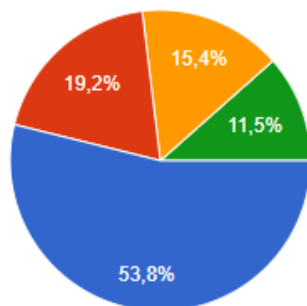
A continuación, se presentan las necesidades de los estudiantes de ingeniería de sistemas, derivadas directamente de la encuesta de necesidades que salieron a su vez de las problemáticas planteadas anteriormente:

Las 7 preguntas planteadas y resultados para la encuesta de necesidades fueron:

1. Para superar la sensación de "no saber por dónde empezar" a practicar, ¿qué tipo de guía te sería más útil?

 Copiar gráfico

26 respuestas



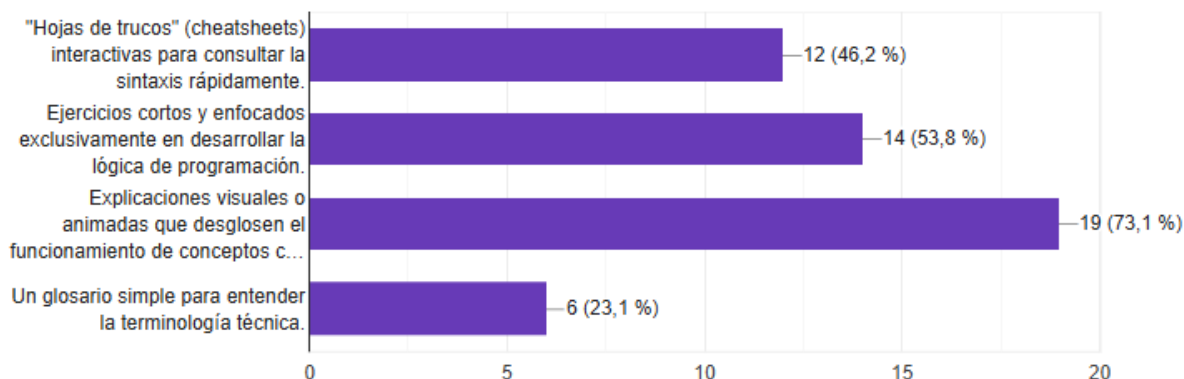
- a) Un plan de estudio estructurado con temas y ejercicios progresivos (un "mapa de ruta").
- b) Una lista de proyectos pequeños y guiados paso a paso.
- c) Un generador de ejercicios aleatorios clasificados por tema y dificultad.
- d) Sugerencias personalizadas basadas en mi progreso y errores anteriores.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

2. Pensando en las dificultades más comunes, ¿qué ayudas específicas te gustaría tener a mano mientras aprendes?

[Copiar gráfico](#)

26 respuestas



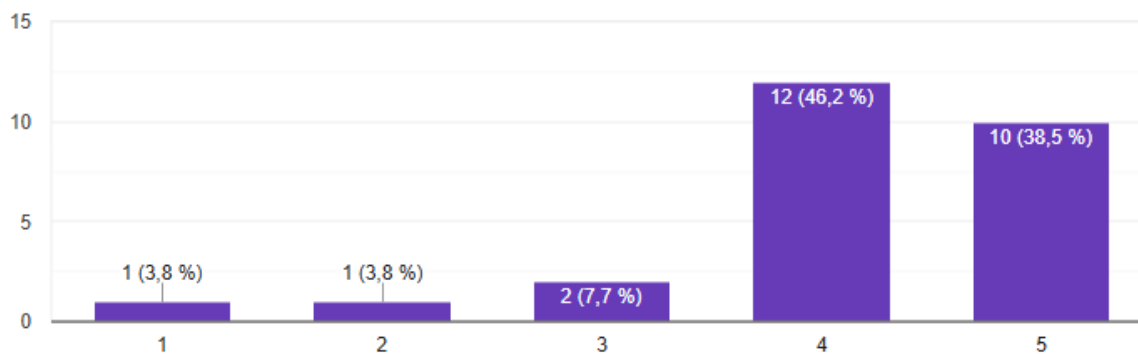
4. En una escala del 1 al 5, ¿qué tan importante sería para ti que una plataforma de aprendizaje te ofrezca ejercicios interactivos con corrección automática e inmediata?

[Copiar gráfico](#)

1 (Poco importante)

5 (Extremadamente importante)

26 respuestas



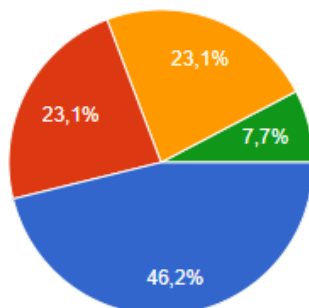


## ISWD523 DISEÑO DE SOFTWARE GR2SW

5. Para mantenerte motivado y combatir la frustración, ¿qué elemento consideras más valioso?

[Copiar gráfico](#)

26 respuestas

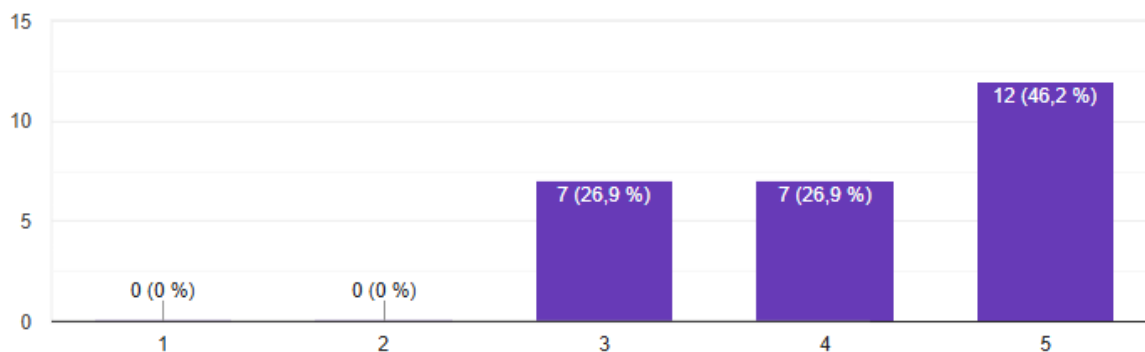


- a) Ver mi progreso de forma visual (barras de avance, mapa de habilidades desbloqueadas).
- b) Recibir recompensas virtuales como insignias o puntos por completar retos (gamificación).
- c) Poder contactar a un mentor o tutor para resolver dudas directamente en l...
- d) Tener un espacio (foro o chat) para colaborar y discutir problemas con otr...

6. En una escala del 1 al 5, ¿qué tan útil te resultaría que el contenido de la aplicación estuviera directamente alineado con el programa de estudios de tu universidad?

[Copiar gráfico](#)

26 respuestas

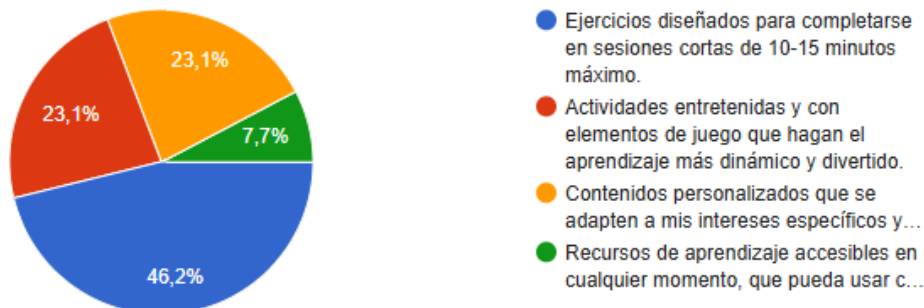


## ISWD523 DISEÑO DE SOFTWARE GR2SW

7. Considerando la falta de tiempo para practicar, ¿qué tipo de características en una actividad de estudio te incentivarían más a dedicarle tiempo extra?

 Copiar gráfico

26 respuestas



Una vez obtenidos los datos de la encuesta, en la cual hubo 26 respuestas, se realizó la interpretación y propuesta de las necesidades generales:

### 1. Necesidades de Contenido y Metodología de Enseñanza

Estas necesidades se centran en cómo se enseña la programación y qué contenido se imparte.

- **Necesidad de Contenido Curricular Sincronizado**

Los estudiantes requieren una herramienta de aprendizaje cuyo contenido esté directamente alineado y sincronizado con el programa de estudios oficial de su universidad.

- **Máxima Relevancia:** El **46.2%** de los estudiantes (12 respuestas) calificó esta característica con un 5, el nivel más alto de utilidad.
- **Alta Aceptación:** Sumando las calificaciones de 4 y 5, un **73.1%** de los encuestados considera que la alineación curricular es muy útil
- **Ausencia de Rechazo:** Es crucial destacar que **ningún estudiante** (0%) consideró que esta funcionalidad fuera poco o nada útil (calificaciones de 1 y 2).

### 2. Necesidades de Práctica y Aplicación

Se refieren a la capacidad de los estudiantes para aplicar lo que aprenden y desarrollar autonomía.

- **Necesidad de estructura en el aprendizaje.**

- El 53.8% de los estudiantes le sería útil un plan de estudio estructurado con temas y ejercicios progresivos.
- El 19.2% prefiere una lista de proyectos pequeños y guiados paso a paso.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

- Al 15.4% le sería útil un generador de ejercicios aleatorios clasificados por tema y dificultad.
- Las sugerencias personalizadas basadas en mi progreso y errores anteriores son preferidas por el 11.5% de los encuestados.
- **Necesidad de actividades interactivas con retroalimentación inmediata para el aprendizaje autónomo.**
  - El 84% de los estudiantes considera importante o extremadamente importante contar con ejercicios interactivos que ofrezcan corrección automática e inmediata, con un 46.2% calificándolo como importante y un 38.5% como extremadamente importante.
  - Solo el 7.6% de los estudiantes considera esta funcionalidad como poco importante.
  - Mientras que el 7.7% mantiene una posición neutral respecto a su valor.
  - Estos datos evidencian que los estudiantes valoran significativamente las herramientas que les permiten practicar de manera independiente con retroalimentación instantánea, lo cual facilita su proceso de aprendizaje.

### 3. Necesidades de Acompañamiento y Retroalimentación

Estas necesidades abordan los aspectos emocionales y de apoyo que impactan el aprendizaje.

- **Necesidad de Retroalimentación visual y tangible sobre su progreso**
  - **Prioridad Alta: Visualización del Progreso (46.2% de las respuestas):** Casi la mitad de los encuestados considera que "Ver mi progreso de forma visual (barras de avance, mapa de habilidades desbloqueadas)" es el elemento más valioso.
  - **Prioridad Media-Alta: Recompensas Virtuales y Soporte Directo (Ambas con 23.1% de las respuestas):** "Recibir recompensas virtuales como insignias o puntos por completar retos (gamificación)" y "Poder contactar a un mentor o tutor para resolver dudas directamente" son valoradas de manera equitativa por un cuarto de los participantes cada una
  - **Prioridad Complementaria: Colaboración entre Pares (7.7% de las respuestas):** "Tener un espacio (foro o chat) para colaborar y discutir problemas con otros" es la opción menos priorizada individualmente. No obstante, el hecho de que un porcentaje de usuarios lo valore indica que un componente comunitario o de apoyo entre pares, aunque no sea el foco principal, puede complementar y enriquecer la experiencia general.

### 4. Necesidades de Recursos de Aprendizaje

Se enfocan en los materiales y herramientas que los estudiantes utilizan para aprender.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

- **Necesidad de herramientas pedagógicas complementarias.**
  - El 73.1 % coincide en que le gustaría tener explicaciones visuales o animadas que desglosen el funcionamiento de conceptos complejos como ayuda.
  - El 53.8% opta por los ejercicios cortos y enfocados exclusivamente en desarrollar la lógica de programación.
  - El 46.2% prefiere las “hojas de trucos” (cheatsheets) interactivas para consultar la sintaxis rápidamente.
  - Y tan sólo el 23.1% cree que sería de ayuda un glosario simple para entender la terminología técnica.
  - Esta situación indica que los materiales y métodos del aula no satisfacen sus necesidades y que existe una falta de materiales educativos claros y accesibles.
- **Necesidad de herramientas de detección y comprensión de errores**
  - El 30.8% requiere de un “traductor” de mensajes de error que explique los problemas en un lenguaje sencillo y claro.
  - El 26.9% necesita herramientas de visualización paso a paso para comprender la ejecución del código.
  - El 23.1% busca la posibilidad de comparar su código con soluciones correctas.
  - El 19.2% prefiere pistas automáticas que identifiquen la ubicación de los errores.
  - Estos resultados indican que los estudiantes enfrentan dificultades significativas para interpretar y resolver errores de programación, requiriendo múltiples enfoques de apoyo que simplifiquen el proceso de identificación y corrección de problemas.

### 5. Necesidad de Aprendizaje Flexible y Altamente Atractivo

Los usuarios tienen una necesidad predominante de **flexibilidad y eficiencia en el tiempo de estudio**, priorizando las actividades que pueden completarse en sesiones cortas. Además, buscan que el aprendizaje sea **entretenido y personalizado**, para que les resulte más atractivo dedicar tiempo adicional.

- **Prioridad Alta: Formato de Estudio Micro-Sesiones (46.2% de las respuestas):** Casi la mitad de los encuestados considera que "Ejercicios diseñados para completarse en sesiones cortas de 10-15 minutos máximo" es la característica principal que los incentivaría a dedicar tiempo extra.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

Esto subraya una fuerte necesidad de contenido que se adapte a agendas apretadas y que pueda ser consumido en "micro-momentos" de tiempo libre, haciendo el estudio más accesible y menos abrumador.

- **Prioridad Media-Alta: Entretenimiento y Gamificación, y Personalización (Ambas con 23.1% de las respuestas):** Las opciones "Actividades entretenidas y con elementos de juego que hagan el aprendizaje más dinámico y divertido" y "Contenidos personalizados que se adapten a mis intereses específicos y carrera profesional" son valoradas igualmente por el 23.1% de los participantes.

Esto indica que, además de la brevedad, la calidad de la experiencia (que sea divertida) y la relevancia del contenido (que sea personalizado) son factores cruciales para captar y mantener el interés de los usuarios para invertir tiempo adicional.

- **Prioridad Complementaria: Accesibilidad en Cualquier Momento (7.7% de las respuestas):** "Recursos de aprendizaje accesibles en cualquier momento, que pueda usar cuando tengan tiempo libre" es la opción menos votada individualmente. Si bien no es el principal motivador para dedicar *tiempo extra*, la accesibilidad es una característica base que soporta el resto de las preferencias, especialmente la de las micro-sesiones.

## Requerimientos

### MoSCoW

The four capitalized letters in the **MoSCoW** prioritization scheme stand for four possible priority classifications for the requirements in a set (IIBA 2009):

- **Must:** The requirement must be satisfied for the solution to be considered a success.
- **Should:** The requirement is important and should be included in the solution if possible, but it's not mandatory to success.
- **Could:** It's a desirable capability, but one that could be deferred or eliminated. Implement it only if time and resources permit.
- **Won't:** This indicates a requirement that will not be implemented at this time but could be included in a future release. [3]

ID RF	Necesidad Relacionada	Requerimiento Funcional	MoSCoW
-------	-----------------------	-------------------------	--------

## ISWD523 DISEÑO DE SOFTWARE GR2SW

<b>RF-1</b>	Contenido curricular sincronizado	El sistema debe permitir la creación y actualización de módulos de contenido alineados con las nuevas actualizaciones	<b>MUST</b>
<b>RF-2</b>	Estructura y personalización del aprendizaje	El sistema debe ofrecer planes de estudio estructurados con niveles progresivos y recomendaciones personalizadas basadas en el desempeño del usuario.	<b>SHOULD</b>
<b>RF-3</b>	Retroalimentación inmediata	El sistema debe proporcionar ejercicios interactivos con corrección automática y retroalimentación instantánea.	<b>MUST</b>
<b>RF-4</b>	Visualización del progreso	El sistema debe mostrar indicadores visuales del progreso, como barras, mapas de habilidades y logros desbloqueados	<b>SHOULD</b>
<b>RF-5</b>	Gamificación y recompensas	El sistema debe incluir un sistema de puntos, insignias y retos para incentivar la participación y el aprendizaje.	<b>SHOULD</b>
<b>RF-6</b>	Gamificación y recompensas	El sistema debe otorgar insignias al usuario por alcanzar logros o hitos predefinidos en su progreso de aprendizaje.	<b>COULD</b>
<b>RF-7</b>	Acceso a mentores y tutorías	El sistema debe permitir a los usuarios contactar mentores o tutores para resolver dudas en tiempo real o diferido.	<b>SHOULD</b>
<b>RF-8</b>	Espacios de colaboración social	El sistema debe incluir foros, chats y grupos para que los usuarios puedan discutir problemas, compartir soluciones y colaborar en proyectos.	<b>SHOULD</b>
<b>RF-9</b>	Materiales visuales y animados	El sistema debe ofrecer explicaciones visuales, animaciones y materiales complementarios para facilitar la comprensión.	<b>MUST</b>
<b>RF-10</b>	Diagnóstico y ayuda en errores	El sistema debe analizar el código ingresado por el usuario y ofrecer traducción de mensajes de error, pistas y comparación con soluciones correctas.	<b>MUST</b>
<b>RF-11</b>	Micro-sesiones	El sistema debe permitir realizar ejercicios y lecciones diseñados para completarse en lapsos de 5 - 10 minutos.	<b>SHOULD</b>

## ISWD523 DISEÑO DE SOFTWARE GR2SW

<b>RF-12</b>	Personalización avanzada	El sistema debe adaptar el contenido y las actividades a los intereses, nivel y carrera profesional del usuario.	<b>SHOULD</b>
--------------	--------------------------	--	---------------

### REQUISITOS NO FUNCIONALES

ID RNF	Tipo	Requisito no funcional	Trazabilidad
<b>RNF-1</b>	Usabilidad	El sistema debe ser fácil de usar para estudiantes sin experiencia previa en plataformas de aprendizaje en línea.	N1 – Contenido y metodología de enseñanza→ RF-1, RF-2, RF-8
<b>RNF-2</b>	Mantenibilidad	El sistema debe permitir actualizaciones del contenido sin afectar la operatividad general.	N1 – Contenido sincronizado→ RF-1
<b>RNF-3</b>	Seguridad	El sistema debe proteger los datos del usuario mediante autenticación segura.	N3 – Acompañamiento y retroalimentación→ RF-6, RF-7
<b>RNF -4</b>	Escalabilidad	El sistema debe poder adaptarse al crecimiento del número de usuarios activos sin que se vea comprometido su rendimiento.	N5 - Necesidad de Aprendizaje Flexible y Altamente Atractivo

## Metodología

Pressman, R. S. (en su obra *"Ingeniería del Software: Un Enfoque Práctico"*) presenta una taxonomía de metodologías de desarrollo, clasificadas en dos categorías principales:

### I. Metodologías Tradicionales (Pressman, pp. 33-40)

1. Modelo en Cascada
2. Modelo de Proceso Incremental
3. Modelos de Proceso Evolutivos:
  - 3.1. Prototipos (modelo evolutivo principal)
  - 3.2. Modelo Espiral (evolutivo y orientado a riesgos)
4. Modelo Concurrente

## ISWD523 DISEÑO DE SOFTWARE GR2SW

5. Modelo Basado en Componentes (notado por Pressman frecuentemente como una técnica transversal).

### II. Metodologías Ágiles (Pressman, pp. 61-75)

1. Programación Extrema (XP)
2. Programación Extrema Industrial (IXP)
3. Desarrollo Adaptativo de Software (DAS)
4. Scrum
5. Método de Desarrollo de Sistemas Dinámicos (MDSD)
6. Cristal (familia de metodologías)
7. Desarrollo Impulsado por Características (DIC)
8. Desarrollo Esbelto de Software (DES)

### III. Justificación de la Elección Metodológica

En primera instancia, se descartan las metodologías tradicionales debido a su rigidez inherente y su incompatibilidad con los plazos acotados del presente proyecto.

La selección se orienta hacia metodologías ágiles, fundamentada en las siguientes características del proyecto:

- Tamaño del equipo: Grande (20 integrantes), sin ser masivo.
- Restricción temporal: Plazo limitado (2 meses).
- Naturaleza del producto: Requiere iteración constante con el usuario y capacidad de respuesta rápida a cambios.
- Arquitectura modular: Presencia de componentes funcionales diferenciados.

Entre las opciones ágiles evaluadas, destacan dos metodologías:

#### 1. Programación Extrema (XP):

##### o Ventajas Identificadas:

- ♣ Enfoque orientado a objetos (Pressman, p. 62).
- ♣ Flexibilidad en la gestión de requisitos: "A medida que avanza el trabajo el cliente puede agregar historias, cambiar el valor de una ya existente, descomponerlas o eliminarlas" (Pressman, p. 63).
- ♣ Diseño adaptable: "El diseño es visto como un artefacto en transición que puede y debe modificarse continuamente" (Pressman, p. 63).



## ISWD523 DISEÑO DE SOFTWARE GR2SW

- ♣ Integración continua: "A medida que las parejas de programadores terminan su trabajo, el código que desarrollan se integra con el trabajo de los demás" (Pressman, p. 63).

- **Desventaja Decisiva:**

- ♣ "Falta de diseño formal. XP desalienta la necesidad del diseño de la arquitectura y, en muchas instancias, sugiere que el diseño de todas las clases debe ser relativamente informal" (Pressman). Esta característica resulta incompatible con los objetivos pedagógicos centrados en el diseño de software dentro del marco académico del proyecto.

## 2. Scrum:

Se propone adoptar Scrum, definido por Pressman como *"un método de desarrollo ágil de software concebido por Jeff Sutherland"* (p. 69). Esta elección se sustenta en su idoneidad para las particularidades del proyecto:

- **Gestión de Equipo Grande (20 personas):**

- ♣ Permite subdivisión en equipos funcionales de 3-6 miembros (ej: 3-4 grupos de 5-6 personas).
- ♣ Roles definidos (Scrum Master, Product Owner, Developers) facilitan la organización.
- ♣ Cada sub-equipo puede enfocarse en un módulo específico (Frontend educativo, Backend de comunidad, Integración).

- **Plazo Ajustado (2 meses):**

- ♣ Sprints cortos (1-2 semanas) posibilitan entregas funcionales incrementales y rápidas.
- ♣ El Product Backlog prioriza funcionalidades esenciales (ej: lecciones básicas + foro de preguntas inicial).

- **Arquitectura Modular:**

- ♣ Se adapta eficazmente a las dos líneas funcionales claras del proyecto:
  1. **Módulo de Aprendizaje:** Lecciones interactivas, ejercicios, seguimiento de progreso.
  2. **Módulo de Comunidad:** Foros, sistema de preguntas/respuestas, votaciones.
- ♣ Las Historias de Usuario independientes facilitan el desarrollo paralelo de módulos.

- **Flexibilidad ante Cambios:**

- ♣ Los Sprint Retrospectives permiten ajustar procesos y prioridades rápidamente ante cambios en requisitos (comunes en contextos académicos).

## Casos de Uso

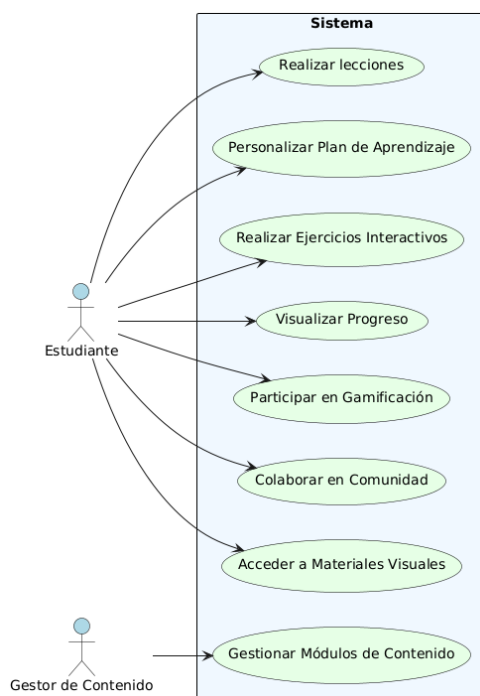


Diagrama Casos de Uso

### Justificación de actores

#### 1. Estudiante

**Rol en el sistema:** Principal beneficiario del sistema interactúa directamente con todas las funcionalidades de aprendizaje.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

**Fundamento:** Los requisitos RF-2 a RF-12 describen actividades y mantenimiento de los módulos de enseñanza, incluyendo textos, ejercicios y elementos multimedia.

### 2. Gestor de contenido

**Rol en el sistema:** Encargado de la creación, actualización y mantenimiento de los módulos de enseñanza, incliyendo textos, ejercicios y elemtnos multimedia.

**Fundamento:** El requisito RF-1 establece la necesidad de un mecanismo para gestionar contenidos acaémicos garantizando que los materiales estén alineados con objetivos curriculares y actualizaciones.

### Formato de Alto Nivel

#### CASO DE USO: Realizar Lecciones

**Actor principal:** Estudiante

**Objetivo en contexto:** Permitir al estudiante acceder a una lección estructurada que contiene teoría, ejemplos y ejercicios prácticos.

**Precondiciones:** El estudiante debe haber iniciado sesión y seleccionado un plan o módulo.

**Disparador:** El estudiante elige una lección desde su panel de estudio.

#### Escenario:

1. El estudiante selecciona una lección disponible.
2. Visualiza el contenido teórico.
3. Interactúa con ejemplos y explicaciones.
4. Pasa al ejercicio práctico asociado.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

5. Finaliza la lección y recibe retroalimentación básica.

### **Excepciones:**

1. Si no hay conexión, se activa modo offline (si descargado previamente).
2. Si el contenido falla al cargar, se muestra mensaje y opción de reintentar.

### **Prioridad:** MUST

**Cuándo estará disponible:** En el primer incremento.

**Frecuencia de uso:** Varias veces al día.

**Canal para el actor:** Interfaz web o app.

**Actores secundarios:** Sistema de evaluación automática.

**Canales para los actores secundarios:** Base de datos de contenido.

### **Aspectos pendientes:**

1. ¿Debe el estudiante tener la opción de repetir una lección completa si ya fue aprobada?
2. ¿Se debe guardar automáticamente el avance al cerrar sesión?
3. ¿Debe haber temporizador o límite de tiempo por lección?
4. ¿Qué criterios definen si una lección está aprobada: solo lectura o resolución de ejercicios?

## CASO DE USO: Personalizar Plan de Aprendizaje

**Actor principal:** Estudiante

**Objetivo en contexto:** Permitir que el estudiante personalice su ruta de aprendizaje según intereses, nivel y desempeño.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

**Precondiciones:** El estudiante debe tener una cuenta activa.

**Disparador:** Accede a la sección “Mi Plan” o responde un cuestionario inicial.

**Escenario:**

1. El sistema analiza el progreso e intereses.
2. Presenta una ruta sugerida.
3. El estudiante ajusta contenido, duración o enfoque.
4. Se guarda la configuración personalizada.

**Excepciones:**

1. Si no hay suficiente información de progreso, se ofrece una ruta estándar.
2. Si el sistema no puede generar un plan, se muestra asistencia guiada.

**Prioridad:** SHOULD

**Cuándo estará disponible:** Segundo incremento.

**Frecuencia de uso:** Cada nuevo ciclo o etapa de aprendizaje.

**Canal para el actor:** Interfaz web/app.

**Actores secundarios:** Motor de personalización.

**Canales para los actores secundarios:** Datos de desempeño, preferencias de usuario.

**Aspectos pendientes:**

1. ¿El estudiante puede elegir entre planes recomendados o crear uno desde cero?
2. ¿Cuánto control tiene el usuario sobre el contenido bloqueado por nivel?
3. ¿Debe incluir recomendaciones automáticas semanales?

## CASO DE USO: Realizar Ejercicios Interactivos

**Actor principal:** Estudiante

**Objetivo en contexto:** Resolver ejercicios prácticos con corrección automática.

**Precondiciones:** Debe estar dentro de una lección activa.

**Disparador:** El estudiante inicia un ejercicio desde la lección.

**Escenario:**

1. Se carga un ejercicio con instrucciones.
2. El estudiante escribe o selecciona la respuesta.
3. El sistema evalúa la respuesta. 4. Se muestra retroalimentación inmediata.

**Excepciones:**

1. Si el código tiene errores, se ofrece diagnóstico.
2. Si hay fallo de red, se guarda en cola y se evalúa más tarde.

**Prioridad:** MUST

**Cuándo estará disponible:** Primer incremento.

**Frecuencia de uso:** Frecuente.

**Canal para el actor:** Editor web / móvil.

**Actores secundarios:** Motor de evaluación.

**Canales para los actores secundarios:** Motor de retroalimentación.

**Aspectos pendientes:**

1. ¿Puede el usuario pausar o guardar la resolución del ejercicio?
2. ¿Se mostrarán pistas o soluciones si el usuario falla repetidamente?

## ISWD523 DISEÑO DE SOFTWARE GR2SW

3. ¿Se deben permitir múltiples intentos por ejercicio?

### CASO DE USO: Visualizar Progreso

**Actor principal:** Estudiante

**Objetivo en contexto:** Ver indicadores visuales del aprendizaje logrado.

**Precondiciones:** Debe haber completado alguna actividad.

**Disparador:** Accede a su perfil o tablero.

**Escenario:**

1. Se cargan estadísticas de progreso.
2. Se presentan en forma de gráficos, barras y logros.
3. El estudiante analiza áreas fuertes y débiles.

**Excepciones:**

1. Si no hay datos suficientes, se muestra mensaje explicativo.

**Prioridad:** SHOULD

**Cuándo estará disponible:** Segundo incremento.

**Frecuencia de uso:** Semanal o después de completar lecciones.

**Canal para el actor:** Dashboard.

**Actores secundarios:** Sistema de análisis de datos.

**Canales para los actores secundarios:** Reportes visuales.

**Aspectos pendientes:**

1. ¿Debe haber alertas automáticas si el progreso baja?
2. ¿Puede el usuario compartir su progreso con otros?

## ISWD523 DISEÑO DE SOFTWARE GR2SW

3. ¿Qué tan detalladas deben ser las estadísticas mostradas?

### CASO DE USO: Participar en Gamificación

**Actor principal:** Estudiante

**Objetivo en contexto:** Motivar el aprendizaje mediante puntos, niveles y recompensas.

**Precondiciones:** Usuario registrado y activo.

**Disparador:** Completa actividades o retos.

**Escenario:**

1. El estudiante gana puntos por actividades.
2. Desbloquea logros o insignias.
3. Compite en rankings o desafíos.

**Excepciones:**

1. Si hay errores al asignar puntos, se muestra mensaje y se reporta al sistema.

**Prioridad:** SHOULD

**Cuándo estará disponible:** Segundo incremento.

**Frecuencia de uso:** Continua.

**Canal para el actor:** Panel de logros.

**Actores secundarios:** Sistema de gamificación.

**Canales para los actores secundarios:** Almacenamiento de puntuaciones.

**Aspectos pendientes:**

1. ¿Debe haber recompensas canjeables o simbólicas?



## ISWD523 DISEÑO DE SOFTWARE GR2SW

2. ¿Se pueden perder puntos por inactividad?
3. ¿Qué tan visibles son los rankings para otros usuarios?

### CASO DE USO: Colaborar en Comunidad

**Actor principal:** Estudiante

**Objetivo en contexto:** Interactuar con otros usuarios en foros o chats.

**Precondiciones:** Cuenta activa y acceso habilitado al foro.

**Disparador:** Accede a la sección “Comunidad”.

**Escenario:**

1. Visualiza temas activos o crea uno nuevo.
2. Publica un comentario o pregunta.
3. Recibe respuestas de otros usuarios.

**Excepciones:**

1. Si hay comportamiento inapropiado, se notifica a moderadores.

**Prioridad:** SHOULD

**Cuándo estará disponible:** Segundo incremento.

**Frecuencia de uso:** Variable.

**Canal para el actor:** Interfaz web/chat.

**Actores secundarios:** Moderador, otros estudiantes.

**Canales para los actores secundarios:** Base de datos de comunidad.

**Aspectos pendientes:**

## ISWD523 DISEÑO DE SOFTWARE GR2SW

1. ¿Debe haber niveles o roles dentro de la comunidad?
2. ¿Se permite el anonimato o solo con identidad confirmada?
3. ¿Qué acciones puede tomar el moderador ante conflictos?

### CASO DE USO: Acceder a Materiales Visuales

**Actor principal:** Estudiante

**Objetivo en contexto:** Ver animaciones, videos o esquemas para apoyar su comprensión.

**Precondiciones:** Estar dentro de una lección o módulo.

**Disparador:** Clic sobre un material visual o recurso adjunto.

**Escenario:**

1. Se carga el recurso multimedia.
2. El estudiante lo visualiza.
3. Puede repetirlo o marcar como favorito.

**Excepciones:**

1. Si el recurso no carga, se ofrece descarga o versión alternativa.

**Prioridad:** MUST

**Cuándo estará disponible:** Primer incremento.

**Frecuencia de uso:** Alta.

**Canal para el actor:** Visor multimedia.

**Actores secundarios:** Sistema de recursos visuales.

**Canales para los actores secundarios:** CDN o almacenamiento local.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

### Aspectos pendientes:

1. ¿Debe permitirse la descarga local para revisión offline?
2. ¿Qué formatos serán soportados (video, gif, 3D)?
3. ¿Se pueden dejar comentarios o notas sobre los recursos?

### CASO DE USO: Gestionar Módulos de Contenido

**Actor principal:** Gestor de contenido

**Objetivo en contexto:** Crear, editar y actualizar módulos temáticos alineados con el plan curricular.

**Precondiciones:** Debe tener permisos de editor o administrador.

**Disparador:** Accede al panel de gestión de contenido.

### Escenario:

1. Selecciona un módulo existente o crea uno nuevo.
2. Agrega lecciones, ejercicios y recursos visuales.
3. Publica o actualiza el módulo.
4. El contenido queda disponible para los estudiantes.

### Excepciones:

1. Si hay errores de validación, se impide guardar.
2. Si se pierden cambios, se muestra mensaje de recuperación.

## ISWD523 DISEÑO DE SOFTWARE GR2SW

**Prioridad:** MUST

**Cuándo estará disponible:** Primer incremento.

**Frecuencia de uso:** Cada vez que hay actualización curricular.

**Canal para el actor:** Panel de edición de contenido.

**Actores secundarios:** Sistema de control de versiones.

**Canales para los actores secundarios:** Base de datos de módulos.

**Aspectos pendientes:**

1. ¿Quién aprueba los cambios antes de que se publiquen?
2. ¿Se puede versionar un módulo y mantener histórico?
3. ¿Qué validaciones automáticas se deben ejecutar antes de guardar?

ID Requerimiento	Requerimiento Funcional	CU1: Realizar Lecciones	CU2: Personalizar Plan	CU3: Ejercicios Interactivos	CU4: Ver Progreso	CU5: Gamificación	CU6: Comunidad	CU7: Materiales Visuales	CU8: Gestionar Módulos
RF-1	Crear y actualizar módulos de contenido								✓
RF-2	Ofrecer planes estructurados y personalizados	✓	✓		✓				
RF-3	Ejercicios interactivos con retroalimentación	✓		✓					
RF-4	Visualización de progreso con indicadores				✓	✓			

## ISWD523 DISEÑO DE SOFTWARE GR2SW

RF-5	Sistema de puntos, insignias y retos			✓		✓			
RF-6	Otorgar insignias por logros					✓			
RF-7	Contacto con mentores o tutores								
RF-8	Espacios de colaboración (foros, chats, proyectos)						✓		
RF-9	Explicaciones visuales y materiales complementarios	✓		✓				✓	✓
RF-10	Diagnóstico y ayuda en errores de código			✓					
RF-11	Micro-sesiones (lecciones rápidas de 5–10 minutos)	✓		✓					
RF-12	Personalización avanzada (intereses, carrera, nivel)		✓		✓	✓			

## Módulos

Este documento describe la modulación del sistema de aprendizaje basado en los casos de uso proporcionados. La modulación divide el sistema en módulos o unidades de trabajo, alineadas con los objetivos de los casos de uso definidos.

### 1. Módulo de Gestión de Contenido

## ISWD523 DISEÑO DE SOFTWARE GR2SW

- **Casos de uso cubiertos:** CU8 – *Gestionar Módulos de Contenido*
  - **Subdominio:** Administración curricular
  - **Entidades:** Modulo, Leccion, Ejercicio, RecursoVisual
  - **Value Objects:** Descripcion, Version, MetadatosCurriculares
  - **Servicios de dominio:** GestorDeModulo, ValidadorDeContenido
  - **Repositorio:** RepositorioModulos
  - **Bounded Context:** Solo editores o administradores operan este contexto.
- 

### 2. Módulo de Lecciones Interactivas

- **Casos de uso cubiertos:** CU1 – *Realizar Lecciones*, CU7 – *Acceder a Materiales Visuales*
  - **Subdominio:** Enseñanza personalizada
  - **Entidades:** LeccionActiva, Estudiante, MaterialVisual
  - **Value Objects:** ContenidoTeorico, EjemploInteractivo, RecursoMultimedia
  - **Servicios de dominio:** ReproductorDeLeccion, ControladorDeSesion
  - **Repositorio:** RepositorioLecciones, RepositorioRecursos
  - **Bounded Context:** Experiencia del estudiante en la lección.
- 

### 3. Módulo de Ejercicios y Evaluación

- **Casos de uso cubiertos:** CU3 – *Realizar Ejercicios Interactivos*
- **Subdominio:** Evaluación práctica
- **Entidades:** Ejercicio, Intento, Resultado
- **Value Objects:** Respuesta, Retroalimentacion, DiagnosticoError
- **Servicios de dominio:** EvaluadorDeEjercicios, GestorDeIntentos

## ISWD523 DISEÑO DE SOFTWARE GR2SW

- **Repositorio:** RepositorioEjercicios, HistorialIntentos
  - **Bounded Context:** Corrección automática y feedback.
- 

### 4. Módulo de Personalización del Aprendizaje

- **Casos de uso cubiertos:** CU2 – *Personalizar Plan de Aprendizaje*
  - **Subdominio:** Ruta de aprendizaje adaptativa
  - **Entidades:** PlanDeAprendizaje, PerfilDelEstudiante
  - **Value Objects:** Intereses, Nivel, Meta
  - **Servicios de dominio:** GeneradorDePlanes, AnalizadorDeProgreso
  - **Repositorio:** RepositorioPlanes
  - **Bounded Context:** Diseño y actualización del camino de aprendizaje.
- 

### 5. Módulo de Progreso y Métricas

- **Casos de uso cubiertos:** CU4 – *Visualizar Progreso*
  - **Subdominio:** Análisis de desempeño
  - **Entidades:** ProgresoEstudiante, Estadística, Logro
  - **Value Objects:** PorcentajeCompletado, Tendencia, HistorialDeAvance
  - **Servicios de dominio:** CalculadorDeProgreso, VisualizadorEstadistico
  - **Repositorio:** RepositorioProgreso
  - **Bounded Context:** Solo lectura y análisis, no modifica datos de lecciones.
- 

### 6. Módulo de Gamificación

- **Casos de uso cubiertos:** CU5 – *Participar en Gamificación*

## ISWD523 DISEÑO DE SOFTWARE GR2SW

- **Subdominio:** Motivación y recompensa
  - **Entidades:** Jugador, Puntaje, Insignia, Desafío
  - **Value Objects:** Nivel, Recompensa, Ranking
  - **Servicios de dominio:** AsignadorDeInsignias, GestorDeRanking
  - **Repositorio:** RepositorioPuntos, HistorialDeLogros
  - **Bounded Context:** No interviene directamente en el contenido educativo.
- 

### 7. Módulo de Comunidad

- **Casos de uso cubiertos:** CU6 – *Colaborar en Comunidad*
  - **Subdominio:** Interacción social
  - **Entidades:** Publicacion, Comentario, Usuario, Moderador
  - **Value Objects:** ContenidoMensaje, RolEnComunidad, Etiqueta
  - **Servicios de dominio:** GestorDeTemas, ModeradorAutomatico
  - **Repositorio:** RepositorioComunidad
  - **Bounded Context:** Separado del aprendizaje individual. Comunica indirectamente con progreso o gamificación.
- 

### 8. Módulo de Infraestructura Compartida (Opcional/Común)

- **Función:** Compartir servicios transversales (autenticación, gestión de archivos, notificaciones)
- **No mapea directamente a un CU**, pero es **necesario** para soportar todos los demás módulos.
- **Servicios:** Autenticador, GestorDeUsuarios, ControladorDeSesiones, GestorOffline
- Puede integrarse como una **capa transversal** más que como un módulo lógico si deseas mantener los 8 principales.



## Referencias

- [1] I. Sommerville, Ingeniería de Software, Madrid: Pearson Educación, 2011.
- [2] Emmy, «Mimo,» 10 Octubre 2023. [En línea]. Available: <https://mimo.org/web/50/section/0>. [Último acceso: 31 Mayo 2025].
- [3] F. P. L. Galicia, «GoDaddy,» 2024 06 14. [En línea]. Available: <https://www.godaddy.com/resources/latam/desarrollo/grasshopper-que-es-como-aprender-programar-con-google>. [Último acceso: 31 Mayo 2025].
- [4] F. P. L. Garcia, «GoDaddy,» 14 Junio 2024. [En línea]. Available: <https://www.godaddy.com/resources/latam/desarrollo/grasshopper-que-es-como-aprender-programar-con-google>. [Último acceso: 31 Mayo 2025].
- [5] C. Collado, «ANDROID4CILL,» 03 Marzo 2023. [En línea]. Available: <https://andro4all.com/aplicaciones/google-va-a-cerrar-a-una-de-sus-aplicaciones-mas-utiles-se-ha-descargado-mas-de-16-millones-de-veces#:~:text=Pero%2C%20como%20parte%20de%20su,ha%20decidido%20abandonar%20el%20proyecto..> [Último acceso: 31 Mayo 2025].
- [6] Enki, «Enki: Learn to code and master technology,» [En línea]. Available: <https://www.enki.com>. [Último acceso: 2025 Mayo 2025].
- [7] «Revista Tino,» [En línea]. Available: <https://revista.jovenclub.cu/sololearn/>. [Último acceso: 03 Junio 2025].
- [8] «proscon.com,» [En línea]. Available: <https://www.proscont.com/ventajas-y-desventajas-sololearn/>. [Último acceso: 04 Junio 2025].

## Anexos:

Anexo 1 : Resultados Encuesta sobre dificultades en el aprendizaje de la programación

[Anexo 1.docx](#)

Anexo 2: Resultados encuesta de necesidades

[Anexo 2 .docx](#)