**Referencia Bibliográfica:**  
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
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):**
   * **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).
     + 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).