

## Conceito: Livro que se Autoconstrói

- **Definição:** Um sistema que identifica as necessidades de conhecimento do usuário e constrói um percurso de aprendizado personalizado, ajustando-se dinamicamente com base nas interações e no progresso.
  - **Inspiração:** Gaps de conhecimento (lacunas detectadas) são preenchidos através de conteúdos relevantes, organizados de forma hierárquica e lógica.
  - **Funcionalidades principais:**
    1. **Mapeamento do Perfil do Usuário:**
      - Identificar conhecimentos prévios, interesses e metas.
      - Diagnóstico inicial através de quizzes, perguntas ou seleção de tópicos.
    2. **Curadoria Adaptativa de Conteúdo:**
      - Escolher e organizar conteúdos relevantes para o usuário com base em lacunas de conhecimento.
      - Hierarquizar informações do básico ao avançado.
    3. **Construção Dinâmica do Livro:**
      - Montar uma estrutura navegável, como capítulos ou módulos, com base nas necessidades mapeadas.
      - Atualizar automaticamente conforme novas necessidades são identificadas.
- 

## Arquitetura e Modelagem

### 1. Estrutura Conceitual

A base do sistema deve ter três níveis principais:

- **Nível 1 (Visão Geral):**
    - Introduz o propósito e oferece uma interface simples para iniciar.
    - **Exemplo:** "Bem-vindo ao Livro que Aprende. Escolha um tema ou faça um teste diagnóstico."
  - **Nível 2 (Geração Adaptativa):**
    - Gera capítulos ou tópicos com base nas lacunas de conhecimento do usuário.
    - **Exemplo:** Um módulo chamado "Conceitos de Grafos" pode ser automaticamente criado se o usuário indicar interesse em modelagem de conhecimento.
  - **Nível 3 (Detalhamento Personalizado):**
    - Oferece conteúdos técnicos, exemplos práticos ou exercícios adaptados ao nível do usuário.
    - **Exemplo:** Explicações detalhadas, exercícios interativos ou visualizações dinâmicas.
- 

### 2. Componentes Técnicos

#### 1. Base de Conhecimento (Knowledge Graph):

- Representa os tópicos e suas relações em um grafo, permitindo a identificação de dependências.

- **Exemplo:**
  - Tópico principal: "Engenharia de Sistemas".
  - Subtópicos relacionados: "Modelagem de Requisitos", "Visualização de Grafos".

## 2. Sistema de Diagnóstico:

- Detecta lacunas no conhecimento através de:
  - Quizzes.
  - Histórico de navegação.
  - Perguntas diretas.

## 3. Motor de Recomendação:

- Utiliza algoritmos (como **Content-Based Filtering** ou **Graph Traversal**) para sugerir conteúdos específicos com base no perfil do usuário.
- **Exemplo:** Se o usuário precisar de "Conceitos Básicos de IA", o sistema sugere capítulos relacionados.

## 4. Gerador de Estrutura Dinâmica:

- Constrói capítulos e seções do livro com base nos dados do usuário.
- **Exemplo:** Gera um índice personalizado como:
  - Capítulo 1: Introdução ao Tema.
  - Capítulo 2: Fundamentos Básicos.
  - Capítulo 3: Casos Práticos.

## 5. Interface Interativa:

- Exibe o conteúdo gerado em uma interface amigável, responsiva e interativa.
- 

## Processo de Construção

### 1. Entrada do Usuário:

- O usuário inicia o sistema informando preferências, interesses ou respondendo perguntas.

### 2. Análise de Necessidades:

- O sistema cruza as informações do usuário com o grafo de conhecimento.

### 3. Construção do Conteúdo:

- Seleção de capítulos, ordenação e organização personalizada.

### 4. Exibição do Livro:

- O usuário recebe um "livro vivo", que pode ser atualizado conforme novos interesses surgem.
- 

## Exemplo Prático

### Cenário:

Um usuário deseja aprender sobre "Visualização de Dados" e responde que tem pouco conhecimento em "Estatística".

### 1. Diagnóstico:

- Lacunas identificadas:
    - Estatística Básica.
    - Fundamentos de Visualização.
2. **Geração de Estrutura:**
    - Capítulo 1: Introdução à Estatística.
    - Capítulo 2: Visualização de Dados Básica.
    - Capítulo 3: Ferramentas Avançadas de Visualização.
  3. **Navegação Dinâmica:**
    - O sistema cria links entre os capítulos, permitindo navegação fácil.
    - Exercícios interativos são adicionados para reforçar lacunas específicas.
- 

## Tecnologias para Implementação

1. **Base de Conhecimento:**
    - Usar um grafo de conhecimento (ex.: Neo4j, RDF, ou estrutura JSON).
  2. **Sistema de Diagnóstico:**
    - Algoritmos de aprendizado supervisionado ou quizzes com pontuação.
  3. **Motor de Recomendação:**
    - Algoritmos baseados em grafos ou sistemas de filtragem.
  4. **Interface Dinâmica:**
    - Frameworks como **React** ou **Vue.js** para criar um front-end interativo.
- 

## Exemplo de Navegação

1. **Home (Nível 1):** Apresenta o sistema e permite diagnóstico.
2. **Conceitos (Nível 2):** Mostra os capítulos sugeridos baseados em lacunas.
3. **Capítulos Gerados (Nível 3):** Traz detalhes técnicos e práticos