



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

NELSON DIAS DE MEDEIROS NETO

**BIGMEMO – PLATAFORMA WEB DE MEMORIZAÇÃO COM REPETIÇÃO
ESPAÇADA E CLASSIFICAÇÃO AUTOMÁTICA DE DIFICULDADE POR IA**

TERESINA-PI
2026

NELSON DIAS DE MEDEIROS NETO

BIGMEMO – PLATAFORMA WEB DE MEMORIZAÇÃO COM REPETIÇÃO ESPAÇADA
E CLASSIFICAÇÃO AUTOMÁTICA DE DIFICULDADE POR IA

Relatório Técnico de Software apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí – IFPI, Campus Teresina Central, como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof^ª. Dr^ª. Aline Montenegro Leal

NELSON DIAS DE MEDEIROS NETO

BIGMEMO – PLATAFORMA WEB DE MEMORIZAÇÃO COM REPETIÇÃO ESPAÇADA
E CLASSIFICAÇÃO AUTOMÁTICA DE DIFICULDADE POR IA

Relatório Técnico de Software apresentado ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí – IFPI, Campus Teresina Central, como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Aprovado em: 17 de junho de 2026.

BANCA EXAMINADORA:

Prof^a. Dr^a. Aline Montenegro Leal

Instituto Federal de Educação, Ciência e Tecnologia do Piauí -
IFPI

Prof^a. Dr^a. Nádia Mendes dos Santos

Instituto Federal de Educação, Ciência e Tecnologia do Piauí -
IFPI

Prof. Dr. Ricardo Martins Ramos

Instituto Federal de Educação, Ciência e Tecnologia do Piauí -
IFPI

TERESINA-PI
2026

Dedico este trabalho não apenas como conclusão de um curso, mas como símbolo de persistência, aprendizado e evolução pessoal.

AGRADECIMENTOS

Agradeço ao IFPI, aos professores e colegas que fizeram parte dessa trajetória acadêmica e contribuíram para meu crescimento pessoal e profissional.

“Aprender não é memorizar por um instante, mas construir conhecimento capaz de permanecer ao longo do tempo.”

Nelson Dias De Medeiros Neto

RESUMO

A retenção de informação a longo prazo é um dos maiores desafios para quem estuda. Experimentos clássicos em psicologia cognitiva mostraram que a memória humana decai exponencialmente em horas ou dias após a aquisição de um novo conhecimento, na ausência de revisão ativa. Técnicas como a repetição espaçada e a prática de recuperação são respostas eficazes a esse problema, com forte respaldo empírico, mas aplicá-las manualmente é inviável para quem precisa gerenciar centenas de conteúdos simultaneamente. Este trabalho apresenta o BigMemo, uma plataforma web criada para auxiliar estudantes e autodidatas a organizar, revisar e reter conteúdo de forma sistemática. O sistema implementa o algoritmo SuperMemo 2 com escala tríade de dificuldade, gera flashcards de forma automatizada por meio de modelos de linguagem de grande escala e disponibiliza um painel de estatísticas de progresso. Duas funcionalidades diferenciam o BigMemo das demais plataformas SRS analisadas: a classificação automática de dificuldade por IA, que avalia semanticamente a resposta escrita pelo usuário e mitiga vieses metacognitivos próprios da autoavaliação; e o índice de retenção estimada por baralho, derivado da curva do esquecimento e atualizado em tempo real, que torna visível o percentual de conhecimento ainda retido pelo estudante. Em conjunto, essas funcionalidades posicionam o BigMemo como alternativa diferenciada entre as plataformas SRS analisadas.

Palavras-chave: repetição espaçada; flashcards; SuperMemo 2; classificação automática de dificuldade por IA; curva do esquecimento.

ABSTRACT

Long-term retention of information is one of the greatest challenges for those who study. Classic experiments in cognitive psychology showed that human memory decays exponentially within hours or days after the acquisition of new knowledge, in the absence of active review. Techniques such as spaced repetition and retrieval practice are effective responses to this problem, with strong empirical support, but applying them manually is unfeasible for anyone managing hundreds of items simultaneously. This work presents BigMemo, a web platform created to help students and autodidacts organize, review, and retain content systematically. The system implements the SuperMemo 2 algorithm with a three-level difficulty scale, generates flashcards automatically through large language models, and provides a progress statistics panel. Two features differentiate BigMemo from the other SRS platforms analyzed: automatic difficulty classification by AI, which semantically evaluates the user's written answer and mitigates the metacognitive biases inherent to self-assessment; and the estimated retention index per deck, derived from the forgetting curve and updated in real time, which makes visible the percentage of knowledge the student still retains. Together, these features position BigMemo as a differentiated alternative among the SRS platforms analyzed.

Keywords: spaced repetition; flashcards; SuperMemo 2; AI-based automatic difficulty classification; forgetting curve.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de casos de uso do BigMemo	27
Figura 2 – Arquitetura em camadas do BigMemo	29
Figura 3 – Fluxograma da execução do algoritmo SuperMemo 2 adaptado . . .	31
Figura 4 – Tela de login da plataforma	32
Figura 5 – Listagem de baralhos do usuário	33
Figura 6 – Tela de revisão manual com a resposta revelada e a dificuldade selecionada	33
Figura 7 – Revisão assistida por IA: classificação e <i>feedback</i> do modelo de linguagem	34
Figura 8 – Tela de geração de cartões por IA com formulário e histórico de gerações	35
Figura 9 – Painel de estatísticas: índice de retenção, métricas e atividade diária	35

LISTA DE TABELAS

Tabela 1 – Comparativo entre BigMemo e plataformas SRS analisadas	19
---	----

LISTA DE ABREVIATURAS E SIGLAS

ACID	<i>Atomicity, Consistency, Isolation, Durability</i> (Atomicidade, Consistência, Isolamento, Durabilidade)
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicação)
CSV	<i>Comma-Separated Values</i> (Valores Separados por Vírgula)
FSRS	<i>Free Spaced Repetition Scheduler</i>
HTTP	<i>HyperText Transfer Protocol</i>
HTTPS	<i>HyperText Transfer Protocol Secure</i>
IA	Inteligência Artificial
JSON	<i>JavaScript Object Notation</i>
MVC	<i>Model-View-Controller</i>
SM-2	SuperMemo 2 (algoritmo de repetição espaçada)
SQL	<i>Structured Query Language</i>
SRS	<i>Spaced Repetition System</i> (Sistema de Repetição Espaçada)

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.1.1	Objetivo Geral	14
1.1.2	Objetivos Específicos	14
2	TRABALHOS RELACIONADOS	16
2.1	Anki	16
2.2	SuperMemo	17
2.3	Quizlet	17
2.4	RemNote	18
2.5	Vantagens do BigMemo	19
3	TECNOLOGIAS ENVOLVIDAS	21
3.1	TECNOLOGIAS DE <i>FRONTEND</i>	21
3.1.1	Hotwire	21
3.1.2	ViewComponent	21
3.1.3	Tailwind CSS	22
3.1.4	Chart.js	22
3.2	TECNOLOGIAS DE <i>BACKEND</i>	22
3.2.1	Ruby on Rails	22
3.2.2	PostgreSQL	23
3.2.3	Sidekiq	23
3.2.4	dry-monads	23
3.2.5	OpenAI API	24
4	MODELAGEM DO PROJETO	25
4.1	LEVANTAMENTO DE REQUISITOS	25
4.1.1	Requisitos Funcionais	25
4.1.2	Requisitos Não Funcionais	26
4.2	DIAGRAMA DE CASOS DE USO	26
4.2.1	Ator Estudante	28
4.2.2	Ator Visitante	28
4.3	ARQUITETURA DO SISTEMA	28
4.3.1	Visão geral	28
4.3.2	Componentes principais	30
4.3.2.1	Frontend	30

4.3.2.2	Backend	30
4.3.3	Modelagem do domínio	30
4.3.4	Algoritmo SuperMemo 2	30
4.3.5	Considerações sobre Segurança	31
4.4	INTERFACES	31
4.4.1	Telas de Acesso	32
4.4.2	Telas de Gerenciamento de Baralhos	32
4.4.3	Telas de Revisão	33
4.4.4	Telas de Painéis e IA	34
5	CONSIDERAÇÕES FINAIS	36
	REFERÊNCIAS	38
6	DECLARAÇÃO DO AUTOR SOBRE A ENTREGA DO CÓDIGO-FONTE	40
7	DECLARAÇÃO DA COORDENAÇÃO DO CURSO SOBRE A ENTREGA DO CÓDIGO-FONTE	42
	Declaração da Coordenação do Curso sobre a Entrega do Código-Fonte . .	42

1 INTRODUÇÃO

Nunca houve tanta oferta de conhecimento acessível. Universidades publicam cursos abertos, plataformas de certificação atualizam conteúdo todos os dias e bibliotecas digitais oferecem acervos antes restritos ao meio acadêmico. Segundo o Censo da Educação Superior de 2022, o Brasil tinha mais de 9,4 milhões de estudantes matriculados em cursos de graduação (Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP), 2023), aos quais se somam milhões de autodidatas que buscam capacitação contínua em idiomas, programação, concursos e áreas técnicas específicas.

Essa oferta abundante, no entanto, não se converte automaticamente em aprendizagem. O desafio real não está em encontrar o que estudar, e sim em reter o que já foi estudado. A memória humana não opera como um disco rígido em que as informações ficam gravadas e permanecem intactas. Ela é seletiva e sensível à passagem do tempo. Ebbinghaus (1913), em experimentos do final do século XIX, mostrou que uma pessoa esquece cerca de 40% de um conteúdo recém-estudado em 24 horas e mais de 75% em uma semana, na ausência de qualquer revisão.

Esse padrão, conhecido como curva do esquecimento, motivou décadas de pesquisa em psicologia cognitiva. Estudos posteriores o confirmaram em contextos educacionais variados e identificaram técnicas capazes de atenuar seus efeitos (CEPEDA et al., 2006). Duas delas têm respaldo robusto na literatura: a repetição espaçada, que distribui as revisões ao longo do tempo, e a prática de recuperação, que exige do aprendiz o esforço ativo de recordar a informação (ROEDIGER; KARPICKE, 2006; KARPICKE; ROEDIGER, 2008). Embora o respaldo científico seja consolidado, aplicar essas técnicas manualmente é inviável para quem lida com grandes volumes de conteúdo.

Dessa limitação surgiram os sistemas de repetição espaçada assistidos por software, ou SRS (do inglês *Spaced Repetition Systems*). Ferramentas como Anki (ANKI, 2026), SuperMemo (SuperMemo World, 2026), Quizlet (Quizlet Inc., 2026) e RemNote (RemNote Inc., 2026) automatizam o cálculo dos intervalos de revisão e são referência para estudantes que buscam maior eficiência. Ainda assim, todas as plataformas analisadas compartilham uma limitação estrutural: dependem da autoavaliação do usuário para classificar o grau de dificuldade da recordação. Bjork (1994) e, posteriormente, Kruger e Dunning (1999) mostraram que indivíduos são sistematicamente imprecisos ao julgar o próprio conhecimento, tendendo a superestimar o que sabem. Esse viés afeta diretamente um sistema SRS: o usuário que classifica respostas como fáceis em excesso recebe intervalos de revisão maiores do que o adequado e esquece mais cedo. Até pouco tempo, não havia alternativa prática a esse modelo. Os modelos de

linguagem de grande escala tornaram viável a avaliação semântica automatizada de respostas escritas (KASNECI et al., 2023).

Outra lacuna recorrente é a opacidade quanto ao estado de memória do estudante. Nenhuma das ferramentas analisadas mostra ao usuário, de forma quantificada, o quanto do conteúdo de um baralho ele ainda retém. A curva do esquecimento é matematicamente modelável (EBBINGHAUS, 1913; WOZNIAK; GORZELAŃCZYK, 1994), e seus parâmetros já estão presentes nos dados que qualquer sistema SRS armazena. Converter esses dados em um indicador de retenção é um passo natural, mas, até onde foi possível verificar, nenhuma das plataformas concorrentes o implementou nativamente.

Como resposta a essas limitações, o presente trabalho propõe o desenvolvimento do BigMemo, uma plataforma web de memorização por repetição espaçada que reúne, em um único sistema, o algoritmo SuperMemo 2 (WOZNIAK, 1987) adaptado para avaliação tríade de dificuldade, classificação automática de dificuldade por inteligência artificial e um índice de retenção estimada por baralho, derivado da curva do esquecimento (EBBINGHAUS, 1913; WOZNIAK; GORZELAŃCZYK, 1994). Essas duas últimas funcionalidades não encontram equivalente direto nas plataformas analisadas e representam a contribuição central deste trabalho.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver uma plataforma web de memorização por repetição espaçada, baseada no algoritmo SuperMemo 2, com classificação automática de dificuldade e geração de flashcards por inteligência artificial, que ofereça ao usuário um índice de retenção estimada por baralho calculado a partir da curva do esquecimento.

1.1.2 Objetivos Específicos

- Implementar o algoritmo SuperMemo 2 adaptado para uma escala tríade de dificuldade (fácil, médio e difícil), com cálculo automático do intervalo de revisão, do fator de facilidade e da data da próxima sessão de estudo para cada card.
- Implementar a geração automatizada de flashcards a partir de um *prompt* textual enviado a modelos de linguagem de grande escala, com controle pelo usuário sobre a quantidade de cards, o modelo utilizado e os parâmetros de criatividade.
- Implementar a classificação automática de dificuldade por inteligência artificial, permitindo ao usuário digitar sua resposta em texto livre e receber uma

avaliação semântica objetiva do desempenho, acompanhada de um *feedback* explicativo.

- Implementar o índice de retenção estimada por baralho, calculado em tempo real a partir da modelagem exponencial da curva do esquecimento.
- Implementar um painel de estatísticas de progresso que apresente ao usuário métricas como atividade diária, distribuição de dificuldade, maturidade dos cards e tendência de desempenho semanal.

2 TRABALHOS RELACIONADOS

Este capítulo analisa quatro plataformas de repetição espaçada estabelecidas no mercado (Anki, SuperMemo, Quizlet e RemNote), com foco em seus pontos fortes, limitações e na relação de cada uma com a proposta do BigMemo. A seleção privilegiou sistemas com base de usuários expressiva, documentação pública disponível e abordagens distintas ao problema da repetição espaçada. A última seção sintetiza as vantagens do BigMemo frente ao panorama levantado.

2.1 ANKI

O Anki (ANKI, 2026) é o sistema SRS de código aberto mais utilizado no mundo. Lançado em 2006 por Damien Elmes, conta com comunidade ativa de desenvolvedores, vasta biblioteca de baralhos públicos compartilhados e aplicativos nativos para *desktop* e dispositivos móveis. Sua arquitetura de agendamento é baseada em uma versão estendida do SM-2 e, mais recentemente, no algoritmo FSRS, desenvolvido pela comunidade para superar limitações do SM-2 em precisão preditiva.

Os principais pontos positivos do Anki são:

- **Robustez do algoritmo:** implementa o SM-2 estendido e, opcionalmente, o FSRS, com agendamento adaptativo por *card*.
- **Suporte a mídia rica:** permite imagens, áudio, vídeo e fórmulas em LaTeX nos *flashcards*.
- **Ecossistema de *plugins*:** ampla coleção de extensões mantidas pela comunidade.
- **Disponibilidade multiplataforma:** aplicações nativas para Windows, macOS, Linux, Android e iOS, com sincronização via AnkiWeb e operação *offline*.

Apesar dos pontos positivos, o Anki apresenta algumas limitações:

- **Interface datada:** a curva de aprendizado é elevada, sobretudo para usuários não técnicos (ANKI, 2026).
- **Ausência de geração de *cards* por IA nativa:** existem *plugins* de terceiros para integração com LLMs, mas não são oficiais, carecem de integração ao fluxo principal e não oferecem rastreabilidade por item gerado.
- **Dependência total da autoavaliação:** o sistema não dispõe de mecanismo para mitigar o viés metacognitivo na classificação de dificuldade, alimentando o algoritmo apenas com o julgamento subjetivo do usuário.

- **Ausência de índice de retenção estimada:** oferece estatísticas históricas detalhadas, mas nenhuma métrica que mostre, em tempo real, o percentual estimado de conhecimento atualmente retido em cada baralho.

2.2 SUPERMEMO

O SuperMemo (SuperMemo World, 2026) é o sistema pioneiro na implementação computacional da repetição espaçada. Desenvolvido por Piotr Wozniak desde meados da década de 1980, foi a plataforma que formalizou o algoritmo SM-2 (WOZNIAK, 1987) e deu origem a todas as suas versões posteriores: SM-5, SM-8, SM-17 e, indiretamente, o FSRs moderno. Oferece recursos avançados como análise de *forgetting index*, múltiplos modos de revisão, suporte a conteúdo hipertextual estruturado e ferramentas de análise de desempenho a longo prazo.

Os principais pontos positivos do SuperMemo são:

- **Maturidade técnica:** base algorítmica sólida, documentada publicamente e referenciada em trabalhos acadêmicos sobre repetição espaçada.
- **Personalização avançada:** oferece nível de controle raramente encontrado em plataformas concorrentes, permitindo ajustes finos no algoritmo de agendamento.
- **Análise detalhada de desempenho:** disponibiliza ferramentas como *forgetting index* e múltiplos modos de revisão voltados a estudos de longo prazo.

Entre as limitações do SuperMemo, destacam-se:

- **Aplicação proprietária e paga:** é uma aplicação Windows fechada, sem versão web moderna com experiência equivalente à do *desktop*.
- **Interface complexa:** foi desenhada para usuários avançados, o que cria barreira de entrada significativa para estudantes casuais.
- **Ausência de integração com IA:** não oferece geração de conteúdo nem avaliação semântica de respostas por modelos de linguagem.
- **Ausência de indicador interpretável de retenção:** as métricas são abundantes, mas voltadas a analistas do próprio sistema, sem um índice de retenção estimada por baralho de fácil interpretação.

2.3 QUIZLET

O Quizlet (Quizlet Inc., 2026), lançado em 2005 por Andrew Sutherland, é uma das plataformas de flashcards mais populares do mundo. Conta com mais de 60 milhões de usuários ativos mensais, biblioteca enorme de baralhos públicos criados pela comunidade e múltiplos modos de estudo (*flashcards*, *match*, *learn*, *test* e *write*). Nos

planos pagos, oferece funcionalidades de IA para geração e explicação de conteúdo, integradas a um assistente chamado Quizlet Q-Chat.

Os principais pontos positivos do Quizlet são:

- **Experiência do usuário:** interface moderna e intuitiva que funciona igualmente bem em navegador e em aplicativos móveis nativos.
- **Componente social:** oferece compartilhamento de baralhos, *rankings* e integração com turmas, atraindo especialmente estudantes do ensino médio e universitário.
- **Múltiplos modos de estudo:** disponibiliza variantes como *flashcards*, *match*, *learn*, *test* e *write*, atendendo a diferentes estilos de aprendizagem.

Por outro lado, o Quizlet apresenta limitações relevantes:

- **Ausência de algoritmo SRS rigoroso:** o modo “Learn” usa lógica adaptativa simplificada, sem cálculo individualizado de intervalos com base em fator de facilidade e histórico de revisões (Quizlet Inc., 2026), o que compromete a efetividade cognitiva a longo prazo.
- **Funcionalidades de IA restritas ao plano pago:** além disso, não contemplam avaliação semântica de respostas para classificação automática de dificuldade.
- **Ausência de indicador de retenção por baralho:** a plataforma não disponibiliza métrica quantitativa do conhecimento retido pelo usuário.

2.4 REMNOTE

O RemNote (RemNote Inc., 2026) é uma plataforma que combina anotações estruturadas (*note-taking*) com repetição espaçada integrada ao fluxo de escrita. Lançada em 2020, propõe um modelo em que o usuário cria notas em um formato proprietário (*rems*, com suporte a hierarquias, referências bidirecionais e *portals*) e pode marcar qualquer fragmento dessas notas como flashcard, que é então agendado pelo SRS interno. O algoritmo de agendamento é uma adaptação do SM-2 ao contexto de notas interconectadas.

Os principais pontos positivos do RemNote são:

- **Integração orgânica entre estudo e anotação:** permite transformar fragmentos das próprias notas em *flashcards* sem necessidade de recriação em outra ferramenta.
- **Hierarquia e referências bidirecionais:** estrutura de *rems* favorece a conexão entre conceitos, especialmente útil em áreas como medicina e direito.
- **Algoritmo SM-2 adaptado:** mantém a base teórica consagrada da repetição espaçada no contexto de notas interconectadas.

As limitações do RemNote são, em boa medida, o reverso da sua virtude:

- **Dependência do sistema de notas proprietário:** para usufruir plenamente do SRS integrado, o usuário precisa aprender e adotar o formato de anotações da plataforma, criando uma curva de aprendizado adicional.
- **Geração de *cards* por IA limitada:** não oferece controle granular sobre modelo, temperatura e quantidade, como existe em plataformas dedicadas.
- **Ausência de classificação automática de dificuldade:** não implementa avaliação semântica de respostas por IA.
- **Ausência de índice de retenção estimada:** não disponibiliza métrica derivada da curva do esquecimento que mostre, em tempo real, o quanto do conteúdo está retido.

2.5 VANTAGENS DO BIGMEMO

A comparação apresentada nas seções anteriores indica que cada uma das plataformas analisadas tem forças reconhecidas, mas compartilha lacunas relevantes. A Tabela 1 sintetiza os principais critérios considerados.

Tabela 1 – Comparativo entre BigMemo e plataformas SRS analisadas

Critério	BigMemo	Anki	SuperMemo	Quizlet	RemNote
Algoritmo de agendamento	SM-2 adaptado	SM-2 / FSRs	SM-2 / SM-17	Adaptativo simples	SM-2 adaptado
Geração nativa por IA	Sim	Não (plugins)	Não	Sim (plano pago)	Limitada
Classificação de dificuldade por IA	Sim (semântica)	Não	Não	Não	Não
Índice de retenção estimada	Sim (Ebbinghaus)	Não	Não	Não	Não
Plataforma principal	Web	Desktop / Mobile	Desktop	Web / Mobile	Web
Código aberto	Sim	Sim	Não	Não	Não
Import / Export CSV	Sim (nativo)	Sim (próprios)	Não padronizado	Parcial	Não padronizado
Curva de aprendizado	Baixa	Alta	Alta	Baixa	Média

Fonte: Elaborado pelo autor com base em (WOZNIAK, 1987; ANKI, 2026; Quizlet Inc., 2026; SuperMemo World, 2026; RemNote Inc., 2026).

Diferente das plataformas apresentadas, o BigMemo reúne em uma única aplicação web três elementos hoje dispersos ou ausentes no ecossistema: um algoritmo SRS fiel ao SM-2, funcionalidades modernas de IA integradas ao fluxo principal e métricas que tornam o estado de memória do usuário visível e compreensível.

Duas funcionalidades específicas não encontram equivalente direto em nenhuma das plataformas analisadas:

- **Classificação automática de dificuldade por IA:** substitui ou complementa a autoavaliação subjetiva do usuário por uma avaliação semântica objetiva da resposta escrita. Essa abordagem atende ao viés metacognitivo descrito por Bjork (1994) e por Kruger e Dunning (1999), e só se tornou viável com a maturação recente dos modelos de linguagem de grande escala (KASNECI et al., 2023).
- **Índice de retenção estimada por baralho:** aplica a fórmula de decaimento exponencial derivada da curva do esquecimento de Ebbinghaus (1913) aos dados do SM-2 de cada *card*, produzindo em tempo real uma estimativa do conhecimento retido pelo usuário. Embora os dados necessários para esse cálculo estejam presentes em qualquer sistema SRS, nenhuma das plataformas concorrentes apresenta esse indicador ao usuário de maneira nativa.

O BigMemo não pretende substituir ferramentas consolidadas como o Anki em seus nichos específicos, como estudo com mídia rica, comunidades de baralhos compartilhados e uso *offline* intensivo. A proposta é oferecer uma alternativa voltada a estudantes que priorizam a consolidação de longo prazo e se beneficiam de um ambiente web simples, com suporte moderno a IA e maior transparência sobre o próprio progresso.

3 TECNOLOGIAS ENVOLVIDAS

Este capítulo apresenta as tecnologias adotadas no desenvolvimento do Big-Memo, organizadas em duas categorias: tecnologias de *frontend*, responsáveis pela camada de apresentação e interatividade, e tecnologias de *backend*, responsáveis pela lógica de negócio, persistência de dados e integração com serviços externos. A escolha do conjunto privilegiou maturidade, produtividade do desenvolvedor e aderência ao paradigma web moderno, mantendo simplicidade na camada cliente sempre que possível.

3.1 TECNOLOGIAS DE *FRONTEND*

3.1.1 Hotwire

O Hotwire (Basecamp, 2025) é uma abordagem moderna para construção de aplicações web que privilegia o envio de HTML pelo servidor, em vez do tráfego de JSON entre cliente e servidor adotado por *Single Page Applications*. A proposta é simples: o servidor renderiza fragmentos de HTML completos, e o cliente os intercala na página de forma reativa, sem necessidade de bibliotecas JavaScript pesadas.

O Hotwire é composto por duas tecnologias complementares. O **Turbo** é responsável por interceptar navegações e submissões de formulários, atualizar trechos específicos da página por meio de *Turbo Frames* e propagar atualizações em tempo real via *Turbo Streams*, frequentemente disparadas por *broadcasts* originados no servidor. O **Stimulus** é uma biblioteca JavaScript leve que permite anexar comportamento a elementos HTML por meio de atributos `data-controller`, mantendo o JavaScript organizado em controladores pequenos e testáveis.

No BigMemo, foram utilizadas as versões turbo-rails 2.0.7 e stimulus-rails 1.3.3, integradas via *bundler* JavaScript às bibliotecas oficiais @hotwired/turbo-rails 8.0.4 e @hotwired/stimulus 3.1.0. Documentação oficial disponível em <<https://hotwired.dev>>.

3.1.2 ViewComponent

O ViewComponent (GitHub, 2025) é uma biblioteca para Ruby on Rails que permite encapsular fragmentos de interface em classes Ruby reutilizáveis e testáveis. Cada componente é composto por uma classe Ruby (com lógica e atributos) e um *template* ERB associado, mantidos lado a lado em arquivos co-localizados. Diferentemente dos *partials* tradicionais do Rails, os ViewComponents podem ser testados em isolamento, com cobertura unitária análoga à de modelos e serviços.

No BigMemo, foi utilizada a versão 2.83.0, com documentação oficial disponível em <<https://viewcomponent.org>>.

3.1.3 Tailwind CSS

O Tailwind CSS (Tailwind Labs, 2025) é uma biblioteca de estilização baseada em classes utilitárias. Em vez de oferecer componentes pré-prontos como Bootstrap ou Material UI, o Tailwind fornece um vocabulário compacto de classes atômicas que são compostas diretamente no HTML para construir as interfaces. Essa abordagem produz folhas de estilo enxutas, pois apenas as classes efetivamente utilizadas são incluídas no *bundle* final.

No BigMemo, foi utilizada a versão 3 do Tailwind CSS, com documentação oficial disponível em <<https://tailwindcss.com>>.

3.1.4 Chart.js

O Chart.js (Chart.js Contributors, 2025) é uma biblioteca JavaScript de código aberto para renderização de gráficos em elementos `canvas` do HTML. Oferece tipos diversos como barras, linhas, rosca, radar e dispersão, com animações nativas e suporte a responsividade. No BigMemo, o Chart.js é responsável pela renderização dos gráficos do painel de estatísticas, incluindo a atividade diária, a distribuição de dificuldade, a maturidade dos cartões e a tendência de desempenho semanal.

A versão utilizada foi a 4.4.0, com documentação oficial disponível em <<https://www.chartjs.org>>.

3.2 TECNOLOGIAS DE *BACKEND*

3.2.1 Ruby on Rails

O Ruby on Rails (HANSSON et al., 2026) é um *framework* web de código aberto criado em 2004 por David Heinemeier Hansson, escrito na linguagem Ruby. É um dos *frameworks* mais maduros do ecossistema web, com arquitetura MVC (Model-View-Controller), suporte nativo a roteamento, validações, migrações de banco de dados, autenticação, internacionalização, testes e processamento assíncrono. Sua filosofia, expressa nos princípios *Convention over Configuration* e *Don't Repeat Yourself*, prioriza a produtividade do desenvolvedor por meio de convenções sensatas e abstrações de alto nível.

A combinação Ruby on Rails com Hotwire permite construir uma aplicação com experiência próxima à de uma *Single Page Application*, sem adotar a complexidade de um *framework* JavaScript pesado como React, Vue ou Angular. No BigMemo, o Rails atua como o núcleo da aplicação, integrando todas as demais tecnologias do *backend* sob uma única arquitetura coesa.

No BigMemo, foi utilizada a versão 7.1.5.2 do Ruby on Rails, executada sobre Ruby 3.3.0, com documentação oficial disponível em <<https://rubyonrails.org>>.

3.2.2 PostgreSQL

O PostgreSQL (The PostgreSQL Global Development Group, 2025) é um sistema de gerenciamento de banco de dados (SGBD) relacional de código aberto, conhecido por sua robustez, conformidade com padrões SQL e capacidade de lidar com operações complexas e de grande volume. Oferece suporte a tipos de dados avançados, índices especializados, transações ACID (Atomicidade, Consistência, Isolamento e Durabilidade), *ENUMs* nativos e o tipo `jsonb` para armazenamento estruturado de dados semiestruturados.

No BigMemo, o PostgreSQL armazena todas as entidades do domínio: usuários, baralhos, cartões, revisões, credenciais de IA, gerações automatizadas e fluxos de transferência. Os campos sensíveis, como a chave de API das credenciais de IA, são armazenados de forma criptografada por meio do recurso nativo de criptografia do Active Record.

A versão utilizada foi a PostgreSQL 14, com documentação oficial disponível em <<https://www.postgresql.org>>.

3.2.3 Sidekiq

O Sidekiq (PERHAM, 2025) é uma biblioteca Ruby de processamento assíncrono de *jobs*, baseada em filas armazenadas no Redis. Permite que operações de longa duração sejam executadas em processos separados do ciclo de requisição-resposta HTTP, com suporte a tentativas automáticas em caso de falha, controle de concorrência e monitoramento por interface web.

No BigMemo, o Sidekiq é responsável pela execução assíncrona de três operações principais: a geração de cartões por inteligência artificial, a importação de baralhos a partir de planilhas CSV e a exportação de baralhos para o mesmo formato. Em todos os casos, o usuário não precisa aguardar sincronamente o término da operação, e o status é atualizado em tempo real na interface quando o *job* é concluído.

A versão utilizada foi a Sidekiq 7.3.2, apoiada em Redis 6.2.5, com documentação oficial disponível em <<https://sidekiq.org>>.

3.2.4 dry-monads

A biblioteca dry-monads (dry-rb Team, 2025) oferece implementações idiomáticas de mônadas funcionais para a linguagem Ruby. No BigMemo, o tipo central é o `Result`, que materializa o resultado de uma operação como `Success` (caminho de sucesso, com o valor produzido) ou `Failure` (caminho de erro, com o motivo da falha).

Esse modelo permite que a lógica de negócio seja organizada em *operations* pequenas e testáveis, com tratamento explícito dos caminhos de sucesso e falha por meio de *pattern matching*, em vez do uso disperso de exceções e retornos implícitos.

A versão utilizada foi a dry-monads 1.6.0, com documentação oficial disponível em <<https://dry-rb.org/gems/dry-monads>>.

3.2.5 OpenAI API

A OpenAI (OpenAI, 2026) é uma organização dedicada ao desenvolvimento de tecnologias avançadas de inteligência artificial, responsável pela família de modelos GPT. Sua API funciona como uma interface RESTful padronizada que permite integrar modelos de linguagem em aplicações por meio de *endpoints* HTTP autenticados por chave de API. A plataforma oferece capacidades em processamento de texto, geração de *embeddings* e classificação semântica, permitindo consumir modelos sem necessidade de hospedá-los em infraestrutura própria.

No BigMemo, a API da OpenAI sustenta duas funcionalidades centrais. A primeira é a geração automatizada de cartões a partir de um *prompt* textual, com controle pelo usuário sobre quantidade, modelo e parâmetros de criatividade. A segunda é a classificação automática de dificuldade durante a revisão assistida por IA, em que o modelo avalia semanticamente a resposta escrita pelo estudante em comparação com a resposta correta cadastrada no cartão.

A integração foi realizada por meio dos *endpoints* de *Chat Completions* da API REST da OpenAI, com documentação oficial disponível em <<https://platform.openai.com/docs/api-reference/chat>>.

4 MODELAGEM DO PROJETO

4.1 LEVANTAMENTO DE REQUISITOS

Nesta seção são apresentados os requisitos do BigMemo, derivados da análise das quatro plataformas SRS discutidas no Capítulo 2 (Anki, SuperMemo, Quizlet e RemNote), das limitações estruturais identificadas nesses sistemas e da literatura sobre repetição espaçada e metacognição. Os requisitos estão divididos em funcionais, que descrevem o comportamento observável do sistema, e não funcionais, que tratam de atributos de qualidade como desempenho, segurança e usabilidade.

4.1.1 Requisitos Funcionais

- **[RF001]** O sistema deve permitir o cadastro, a autenticação e o gerenciamento de contas dos usuários, com confirmação por e-mail no primeiro acesso e recuperação de senha.
- **[RF002]** O sistema deve permitir que o usuário crie, edite e remova baralhos, registrando nome e descrição de cada um.
- **[RF003]** O sistema deve permitir que o usuário crie, edite e remova cartões em seus baralhos, com pergunta (frente) e resposta (verso) como campos obrigatórios.
- **[RF004]** O sistema deve permitir a importação de baralhos completos a partir de planilhas no formato CSV padronizado.
- **[RF005]** O sistema deve permitir a exportação de baralhos em CSV, preservando a estrutura e o conteúdo dos cartões.
- **[RF006]** O sistema deve oferecer sessões de revisão manual, em que o usuário lê a pergunta, recupera mentalmente a resposta e classifica sua própria dificuldade em uma escala tríade (fácil, médio ou difícil).
- **[RF007]** O sistema deve oferecer sessões de revisão assistida por inteligência artificial, em que o usuário digita sua resposta em texto livre e recebe uma classificação semântica de dificuldade gerada por modelo de linguagem.
- **[RF008]** O sistema deve permitir a geração automatizada de cartões por inteligência artificial a partir de um *prompt* textual, com controle pelo usuário sobre quantidade, modelo e parâmetros de criatividade.
- **[RF009]** O sistema deve calcular, para cada cartão, o intervalo de revisão, o fator de facilidade e a data da próxima sessão de estudo, conforme o algoritmo SuperMemo 2 adaptado para a escala tríade.

- **[RF010]** O sistema deve calcular e exibir, em tempo real, o índice de retenção estimada por baralho, derivado da curva do esquecimento de Ebbinghaus.
- **[RF011]** O sistema deve apresentar um painel de estatísticas por baralho, com métricas de atividade diária, distribuição de dificuldade, maturidade dos cartões e tendência de desempenho semanal.
- **[RF012]** O sistema deve permitir o cadastro e a verificação de credenciais de inteligência artificial, exigidas para o uso das funcionalidades dependentes de modelos de linguagem.

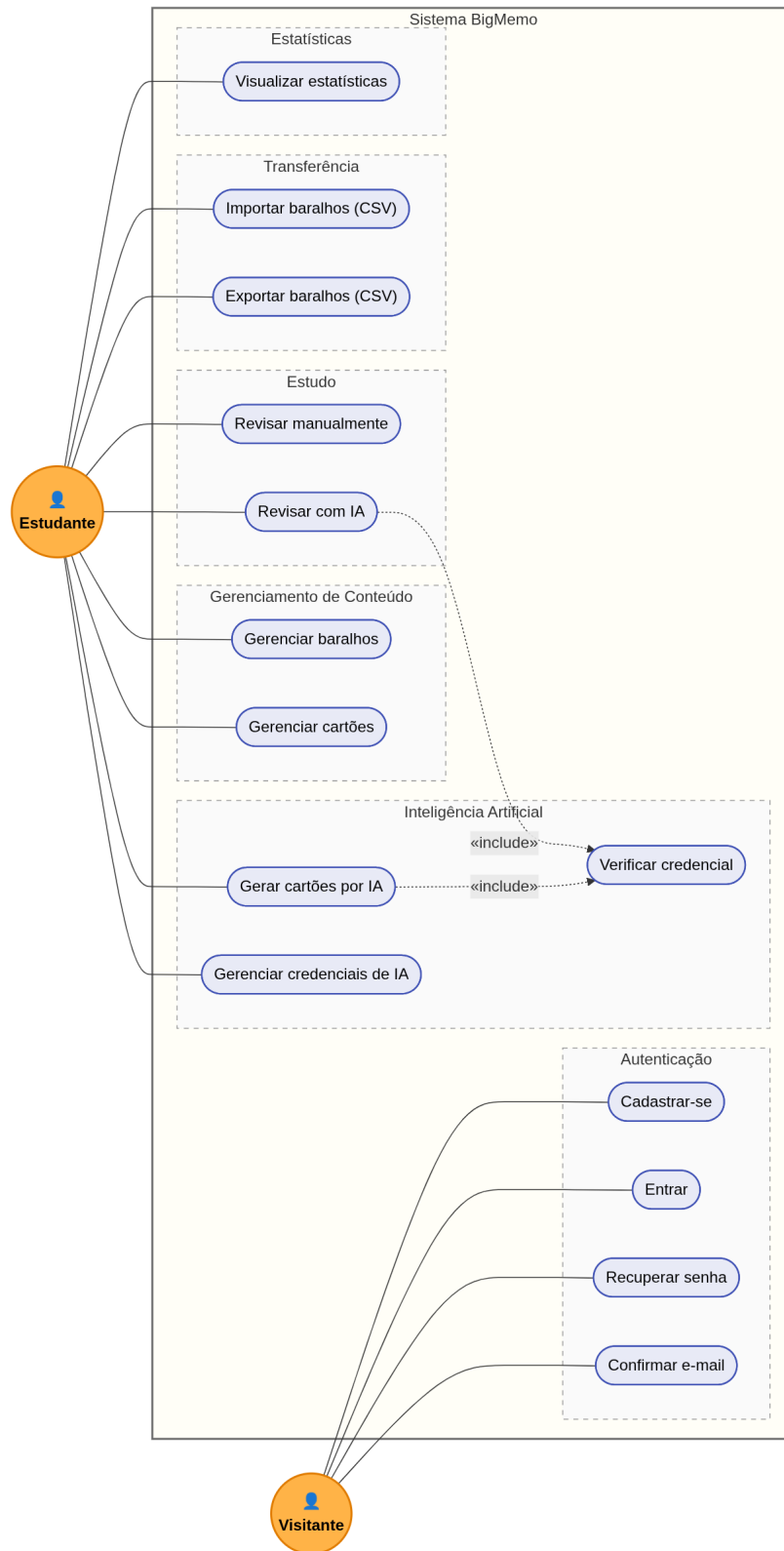
4.1.2 Requisitos Não Funcionais

- **[RNF001]** O sistema deve responder às interações do usuário em tempo inferior a 2 segundos em condições normais de uso.
- **[RNF002]** O sistema deve ser capaz de atender simultaneamente múltiplos acessos de usuários sem degradação perceptível na experiência de uso.
- **[RNF003]** As credenciais sensíveis, como a chave de API do provedor de IA, devem ser armazenadas de forma criptografada no banco de dados.
- **[RNF004]** As operações de longa duração, como geração de cartões por IA e processamento de planilhas CSV, devem ocorrer de forma assíncrona, sem bloquear a interface do usuário.
- **[RNF005]** O sistema deve ser acessível pelos principais navegadores modernos em suas versões suportadas, mantendo comportamento consistente em diferentes resoluções de tela.
- **[RNF006]** O código-fonte deve seguir as boas práticas de desenvolvimento adotadas no ecossistema Ruby on Rails e estar coberto por testes automatizados em três níveis (unitário, integração e sistema).

4.2 DIAGRAMA DE CASOS DE USO

O BigMemo tem um ator principal, o **Estudante**, que representa o usuário autenticado da plataforma, e um ator secundário, o **Visitante**, que interage apenas com as funcionalidades de autenticação antes de obter acesso ao sistema. A Figura 1 apresenta o diagrama de casos de uso do BigMemo.

Figura 1 – Diagrama de casos de uso do BigMemo



Fonte: Elaborado pelo autor.

4.2.1 Ator Estudante

O ator Estudante é o usuário autenticado da plataforma e concentra a maior parte das funcionalidades do sistema. Suas principais interações são:

- **Gerenciar baralhos e cartões:** criar, editar e remover baralhos, bem como adicionar, editar e remover os cartões associados.
- **Importar e exportar baralhos:** enviar planilhas CSV para popular o sistema com baralhos completos e gerar arquivos CSV a partir de baralhos existentes.
- **Revisar cartões:** iniciar sessões de revisão manual, em que classifica a própria dificuldade na escala tríade, ou sessões de revisão assistida por IA, em que digita a resposta em texto livre e recebe uma classificação semântica.
- **Gerar cartões por IA:** solicitar a criação automatizada de cartões a partir de um *prompt* textual, com controle sobre quantidade, modelo e parâmetros.
- **Visualizar estatísticas:** acessar o painel de progresso por baralho, com índice de retenção estimada, métricas resumidas, atividade diária e gráficos analíticos.
- **Gerenciar credenciais de IA:** cadastrar, verificar e remover credenciais de provedor de inteligência artificial.

4.2.2 Ator Visitante

O Visitante é o usuário não autenticado e interage apenas com o subconjunto de funcionalidades necessárias para obter acesso à plataforma:

- Realizar cadastro de nova conta.
- Autenticar-se com e-mail e senha.
- Iniciar o fluxo de recuperação de senha.
- Confirmar a conta por meio do *link* recebido por e-mail.

Os casos de uso mais críticos para as contribuições deste trabalho, *Revisar card com IA* e *Gerar cards por IA*, compartilham o pré-requisito de uma credencial de IA verificada, representado pelo relacionamento de inclusão (*include*) no diagrama da Figura 1.

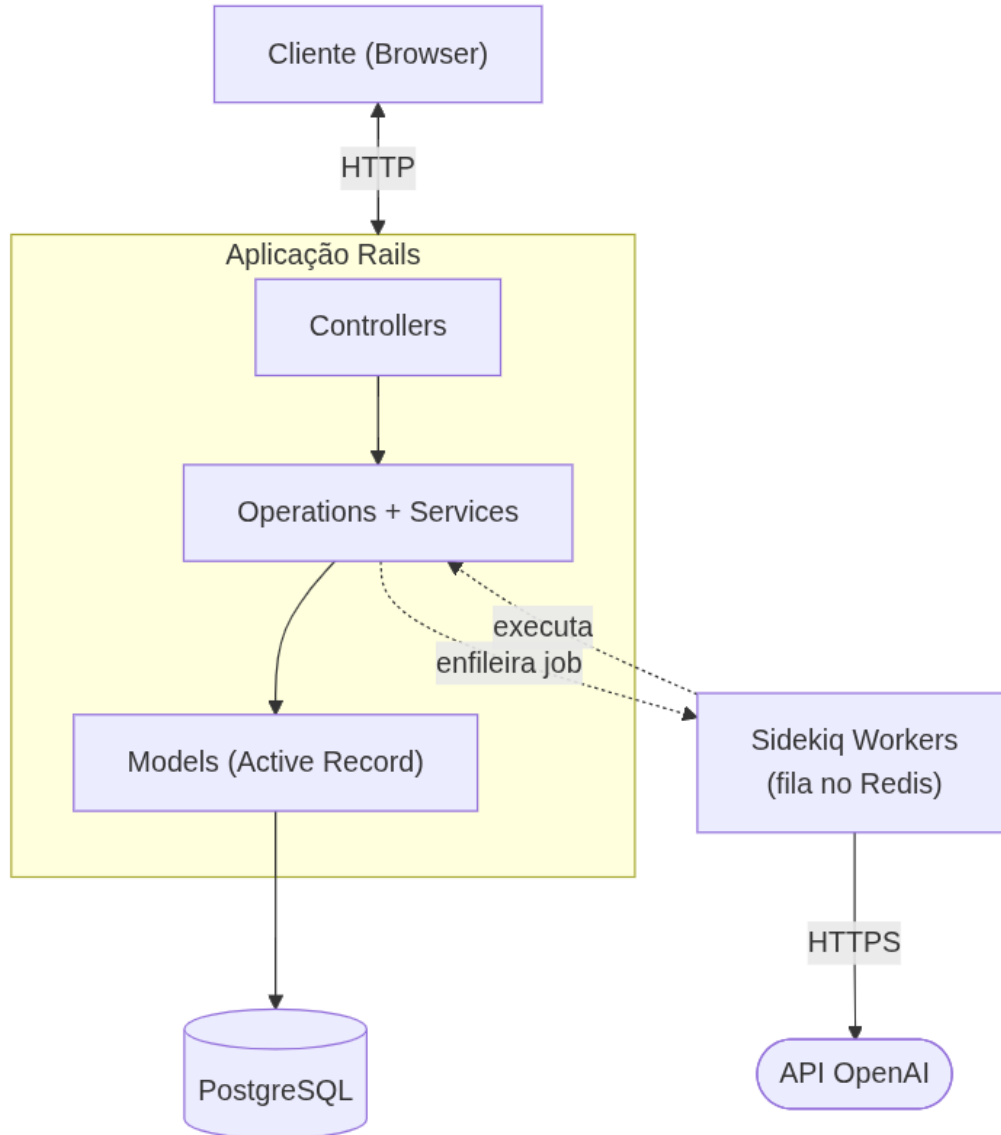
4.3 ARQUITETURA DO SISTEMA

4.3.1 Visão geral

O BigMemo adota uma arquitetura em camadas, na qual cada camada tem responsabilidade bem delimitada e depende apenas das camadas imediatamente inferiores. A Figura 2 apresenta a visão geral da arquitetura, destacando também o

fluxo de processamento assíncrono via Sidekiq e a integração com o provedor externo de IA.

Figura 2 – Arquitetura em camadas do BigMemo



Fonte: Elaborado pelo autor.

O fluxo típico de uma requisição HTTP percorre as camadas de cima para baixo. O *browser* envia uma requisição interceptada pelo Turbo, que a encaminha ao roteador do Rails. O controlador correspondente autentica o usuário, extrai os parâmetros e delega a operação ao componente apropriado da camada de negócio. A operação, implementada com *dry-monads*, retorna um resultado *Success* ou *Failure*, e o controlador usa *pattern matching* para decidir a resposta adequada. A resposta é renderizada por um *ViewComponent* ou por um *template* ERB e retorna ao cliente, frequentemente como um fragmento HTML dentro de um *Turbo Frame* ou *Turbo Stream*, substituindo apenas a parte afetada da página.

4.3.2 Componentes principais

4.3.2.1 Frontend

A camada de *frontend* é composta por *templates* ERB, ViewComponents reutilizáveis e controladores Stimulus que adicionam comportamento dinâmico onde necessário. A interatividade reativa, como a atualização das listas de cartões e dos status de operações assíncronas, é obtida pela combinação de *Turbo Frames* e *Turbo Streams*. A estilização é feita com Tailwind CSS, e os gráficos do painel de estatísticas são renderizados com Chart.js.

4.3.2.2 Backend

A camada de *backend*, implementada em Ruby on Rails, recebe as requisições e orquestra a execução das operações de domínio. Os controladores resolvem as operações por meio de um contêiner de injeção de dependências, e cada operação retorna um `Result` explícito de sucesso ou falha. As operações de longa duração, como geração de cartões por IA e processamento de planilhas CSV, são enfileiradas em *jobs* do Sidekiq e executadas em processos separados, com atualização do *status* em tempo real para o cliente.

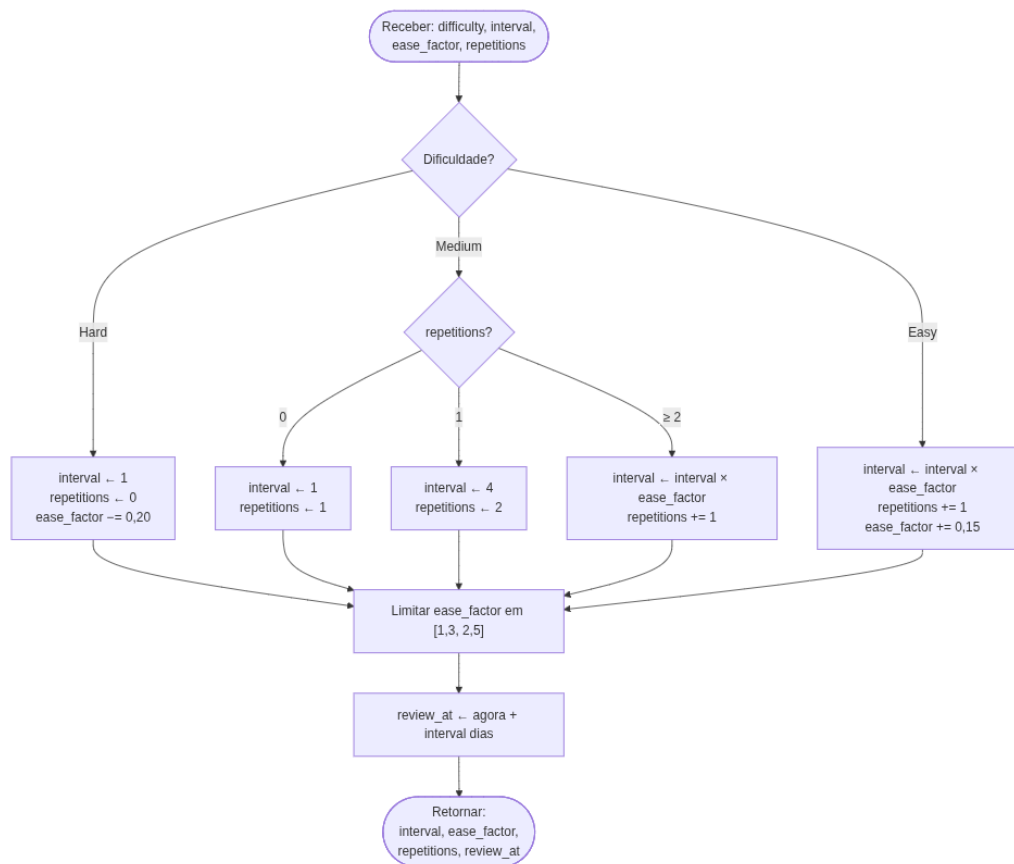
4.3.3 Modelagem do domínio

A modelagem do BigMemo reflete três grandes áreas funcionais. O **domínio de estudo** abriga as entidades centrais para a experiência do usuário (baralhos, cartões e registros de revisão), que materializam o ciclo de repetição espaçada. O **domínio de inteligência artificial** gerencia as credenciais de API dos usuários e as requisições de geração de conteúdo por modelos de linguagem, com rastreabilidade no nível de cada item gerado. O **domínio de transferência** cuida das operações assíncronas de importação e exportação de baralhos em formato CSV, implementadas sobre uma classe abstrata comum que controla o ciclo de vida de cada transferência (estados `queued`, `running`, `success`, `failed` e `error`).

4.3.4 Algoritmo SuperMemo 2

A implementação do SuperMemo 2 no BigMemo é encapsulada em um serviço que recebe quatro parâmetros de entrada (a dificuldade informada pelo usuário ou inferida pela IA, o intervalo atual do cartão, o fator de facilidade atual e o número de repetições consecutivas) e retorna os novos valores desses parâmetros junto com o *timestamp* da próxima revisão. A Figura 3 apresenta o fluxo de execução do serviço, adaptado para uma escala tríade de dificuldade (fácil, médio e difícil), em vez da escala original de seis níveis proposta por Wozniak (1987).

Figura 3 – Fluxograma da execução do algoritmo SuperMemo 2 adaptado



Fonte: Elaborado pelo autor.

4.3.5 Considerações sobre Segurança

A segurança do sistema é implementada em múltiplas camadas. A comunicação entre cliente e servidor é criptografada via HTTPS, protegendo os dados em trânsito. O *backend* implementa autenticação e autorização, garantindo que somente usuários autenticados acessem suas próprias informações. Todas as entradas do usuário são validadas no *backend* antes de serem processadas ou armazenadas, prevenindo injeção de código e ataques similares. As chaves de acesso da API do provedor de IA e as credenciais de banco de dados são armazenadas de forma segura, com criptografia nativa do Active Record para os campos sensíveis.

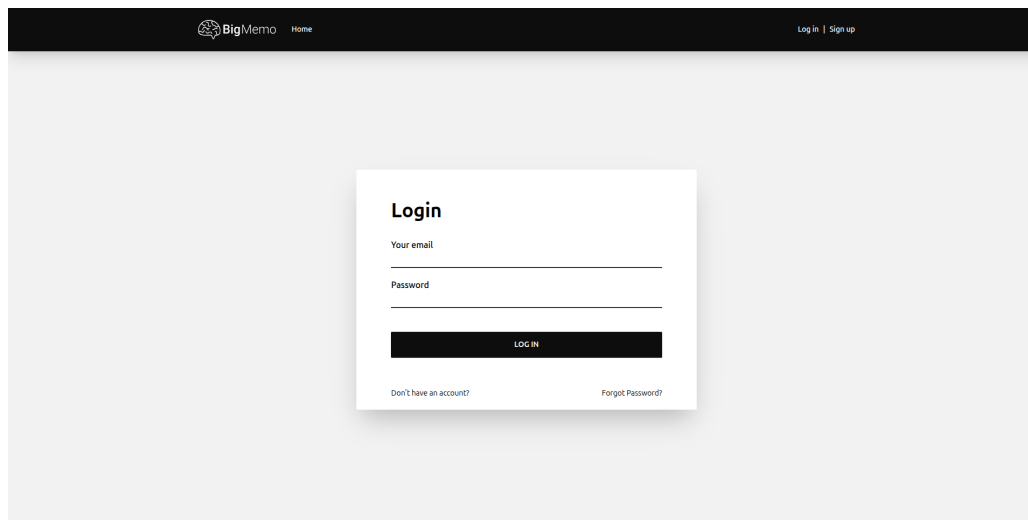
4.4 INTERFACES

Esta seção apresenta as principais interfaces do BigMemo, organizadas pelo fluxo de uso da plataforma. As telas selecionadas para apresentação representam os fluxos mais relevantes do sistema: o acesso inicial, o gerenciamento dos baralhos, as duas modalidades de revisão e os recursos que envolvem inteligência artificial.

4.4.1 Telas de Acesso

A tela de autenticação (Figura 4) é a porta de entrada da plataforma para usuários já cadastrados. Solicita o e-mail e a senha do usuário e oferece atalhos para o cadastro de nova conta e para o fluxo de recuperação de senha. O primeiro acesso à plataforma só é liberado após confirmação por e-mail.

Figura 4 – Tela de login da plataforma

A imagem mostra a interface de login da plataforma BigMemo. No topo, há uma barra preta com o logo 'BigMemo' e o link 'Home' à esquerda, e os links 'Log in' e 'Sign up' à direita. O corpo principal da página é cinza claro. No centro, há um formulário branco com o título 'Login'. Abaixo do título, há dois campos de entrada: 'Your email' e 'Password', cada um com uma linha de texto e uma borda inferior. Abaixo dos campos, há um botão preto com o texto 'LOG IN' em branco. Na base do formulário, há dois links de texto: 'Don't have an account?' e 'Forgot Password?'.

BigMemo Home Log in | Sign up

Login

Your email

Password

LOG IN

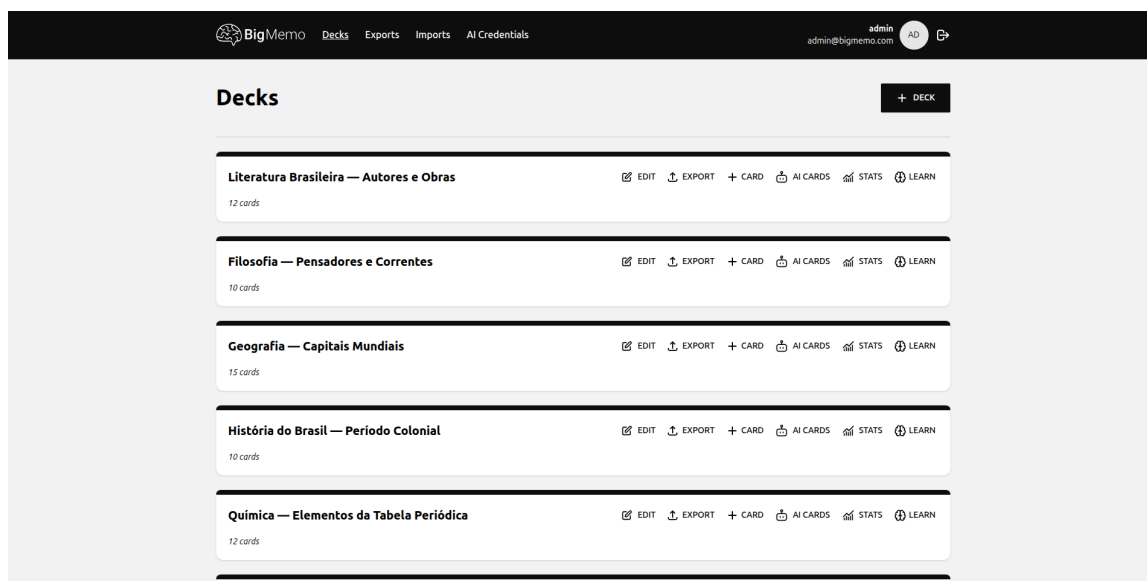
Don't have an account? Forgot Password?

Fonte: Elaborado pelo autor.

4.4.2 Telas de Gerenciamento de Baralhos

A listagem de baralhos (Figura 5) é a tela inicial após a autenticação e funciona como ponto central de navegação da plataforma. A barra superior dá acesso às quatro áreas principais do sistema: baralhos, importações, exportações e credenciais de IA. Cada baralho é exibido em um cartão com seu nome, a quantidade de cartões associados e um conjunto de ações: edição de metadados e cartões, exportação em CSV, criação de novo cartão, geração de cartões por IA, abertura do painel de estatísticas e início de uma sessão de revisão.

Figura 5 – Listagem de baralhos do usuário



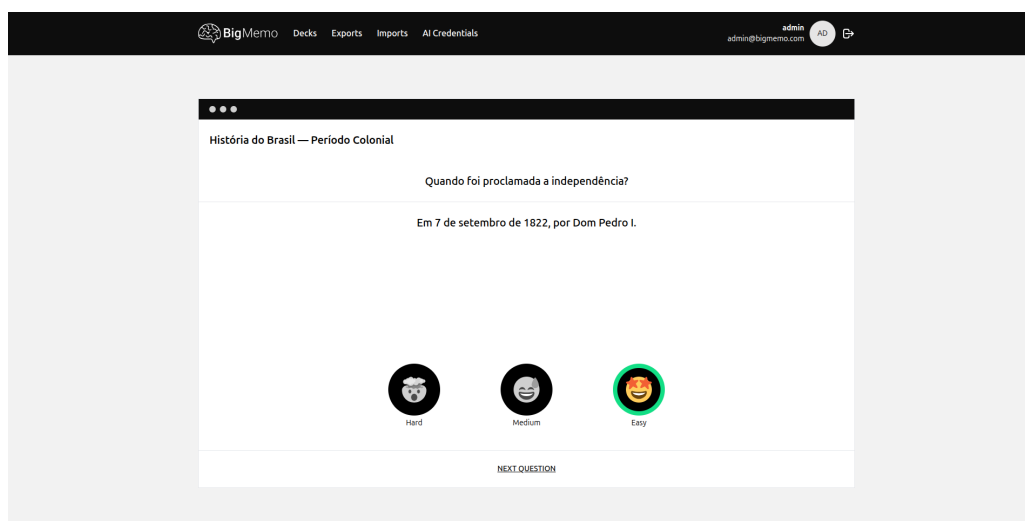
Fonte: Elaborado pelo autor.

4.4.3 Telas de Revisão

A revisão de cartões é o fluxo central da plataforma e está disponível em duas modalidades: manual e assistida por IA.

Na **revisão manual** (Figura 6), o usuário lê a pergunta do cartão e realiza internamente o esforço de recuperação. Ao revelar a resposta, escolhe entre três opções de avaliação de dificuldade, na escala tríade adotada pela plataforma (fácil, médio ou difícil). A escolha alimenta o algoritmo SuperMemo 2, que atualiza o intervalo, o fator de facilidade e a data da próxima revisão do cartão.

Figura 6 – Tela de revisão manual com a resposta revelada e a dificuldade selecionada

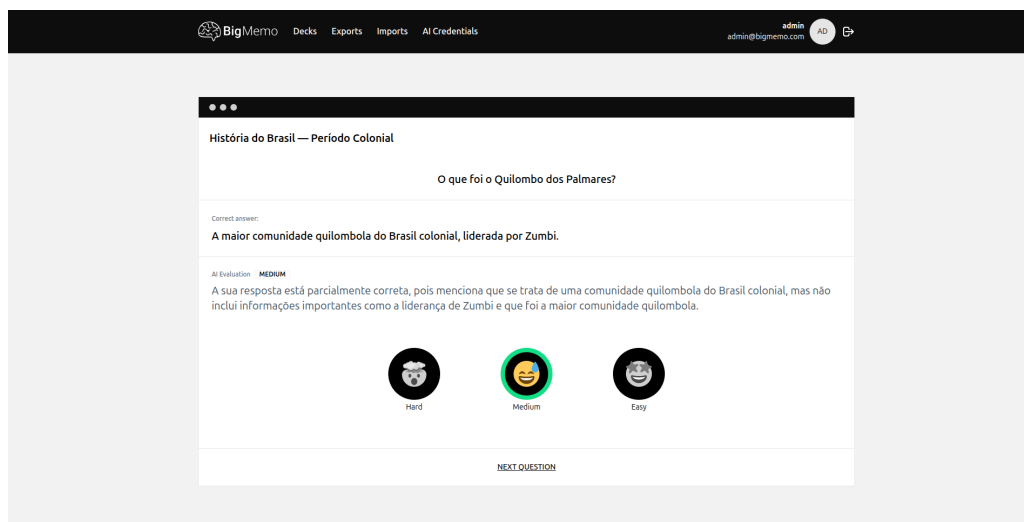


Fonte: Elaborado pelo autor.

Na **revisão assistida por IA** (Figura 7), o fluxo muda no ponto crítico da avali-

ação. Em vez de revelar a resposta e autoavaliar-se, o usuário primeiro digita a sua resposta em texto livre. Após a submissão, o sistema envia ao modelo de linguagem a pergunta, a resposta correta cadastrada no cartão e a resposta escrita pelo usuário, e o modelo retorna uma classificação de dificuldade na mesma escala tríade da revisão manual, acompanhada de uma justificativa textual. A classificação é determinada pela IA e não pode ser ajustada pelo usuário antes da confirmação.

Figura 7 – Revisão assistida por IA: classificação e *feedback* do modelo de linguagem



Fonte: Elaborado pelo autor.

4.4.4 Telas de Painéis e IA

A geração automatizada de cartões por IA (Figura 8) é acessada por ação dedicada na linha de cada baralho. O formulário solicita a credencial de IA a ser utilizada, o modelo de linguagem disponível na credencial selecionada, a quantidade desejada de cartões (limitada a vinte e cinco por execução) e um *prompt* em texto livre que descreve o conteúdo a ser produzido. Abaixo do formulário, a área de histórico lista as gerações anteriores em ordem cronológica decrescente, com data, trecho inicial do *prompt*, quantidade solicitada, quantidade efetivamente criada e *status* da execução.

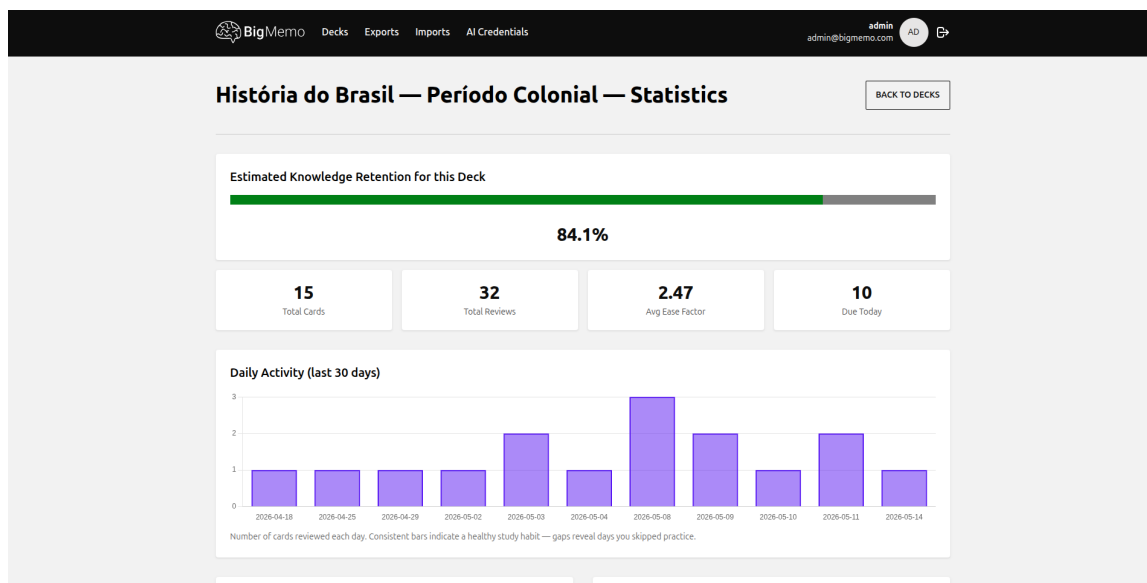
Figura 8 – Tela de geração de cartões por IA com formulário e histórico de gerações

The screenshot shows the 'História do Brasil — Período Colonial — AI Card Generation' page. It features a form to generate cards with AI. The form includes a dropdown for 'AI Credential', a dropdown for 'Model', a text input for 'Prompt' (with an example: 'e.g. Generate flashcards about Ruby on Rails fundamentals'), and a 'Number of cards' input set to 5. There is a '+ GENERATE' button. Below the form is a 'Generation history' section showing a successful generation on May 16, 2026, at 19:06:54, with 5 cards requested and 5 created. A 'BACK TO DECKS' button is in the top right.

Fonte: Elaborado pelo autor.

O painel de estatísticas (Figura 9) é o componente mais informacional da plataforma. Apresenta, em destaque, o índice de retenção estimado do baralho, calculado em tempo real pela fórmula derivada da curva do esquecimento de Ebbinghaus, seguido de quatro métricas resumidas (total de cartões, total de revisões, fator de facilidade médio e cartões com revisão pendente) e do gráfico de atividade diária dos últimos trinta dias. Em uma segunda parte do painel, três visualizações analíticas complementam o conjunto: distribuição de dificuldade, distribuição de maturidade dos cartões e tendência de desempenho das últimas oito semanas.

Figura 9 – Painel de estatísticas: índice de retenção, métricas e atividade diária



Fonte: Elaborado pelo autor.

5 CONSIDERAÇÕES FINAIS

O desenvolvimento do BigMemo permite demonstrar como princípios cognitivos estabelecidos há mais de um século sobre memória, esquecimento e repetição espaçada podem ser colocados em prática em uma aplicação web moderna, combinados a recursos contemporâneos de inteligência artificial. A arquitetura proposta, composta por Ruby on Rails, Hotwire, PostgreSQL, Sidekiq e integração com a API da OpenAI, mostra-se adequada para o processamento das operações de domínio (criação e revisão de cartões, geração automatizada por IA, importação e exportação de baralhos em CSV) e atende aos requisitos definidos para o projeto, oferecendo uma experiência de uso simples e grandes possibilidades de melhorias futuras.

Do ponto de vista técnico, os cinco objetivos específicos estabelecidos no Capítulo 1 foram cumpridos. A implementação do SuperMemo 2 adaptado preserva os princípios centrais do algoritmo original e ajusta intervalos e fator de facilidade conforme o desempenho do usuário. A geração de cartões por IA está integrada ao fluxo de criação de baralhos, com controle granular de parâmetros e rastreamento item a item. O painel de estatísticas oferece ao usuário visibilidade sobre diversas dimensões de seu progresso, incluindo uma métrica inédita entre as plataformas analisadas: o índice de retenção estimada por baralho, calculado em tempo real a partir da curva do esquecimento. A classificação automática de dificuldade por IA, por sua vez, atua sobre o problema do viés metacognitivo descrito na literatura sobre autoavaliação e representa, junto com o índice de retenção, a contribuição central deste trabalho.

Entre as limitações identificadas, destacam-se a dependência de provedor externo de inteligência artificial no modo de revisão assistida, o que implica custo por uso e exige que o estudante cadastre e custeie a própria chave de API; a qualidade variável da classificação, sujeita às características do modelo utilizado e potencialmente inconsistente em respostas ambíguas; e a ausência de suporte a conteúdo multimídia nos cartões, recurso presente em plataformas como Anki e SuperMemo e relevante para domínios como aprendizagem de idiomas e disciplinas anatômicas.

Como trabalhos futuros, diversas oportunidades de evolução são naturais a partir do estado atual. A construção de uma comunidade de compartilhamento de baralhos é, talvez, a linha de maior impacto no curto prazo: o formato CSV padronizado de importação e exportação já serve como contrato de interoperabilidade, e a infraestrutura assíncrona suporta transferências de grandes volumes; a partir dessa base, é viável construir um diretório público com busca por tópico, idioma e popularidade, versionamento, moderação e atribuição de autoria. A expansão do suporte a provedores de IA é outra direção viável, dado que a arquitetura atual já prevê uma interface comum de cliente com métodos de verificação, geração e consulta de modelos. Por fim, a

inclusão de suporte a mídia rica (imagens, áudio e equações matemáticas em LaTeX) seria uma extensão de alto impacto para estudantes de idiomas, medicina e ciências exatas, viabilizada com modificações pontuais sobre a infraestrutura de Active Storage já presente.

O BigMemo não se apresenta como produto acabado. Funciona como ponto de partida estruturado para aplicações que combinem ciência cognitiva e inteligência artificial em favor da aprendizagem. As duas contribuições centrais do trabalho, a classificação automática de dificuldade e o índice de retenção estimada, pretendem contribuir para um debate mais amplo sobre como tornar visível e acionável o que a ciência já sabe sobre memória e esquecimento.

REFERÊNCIAS

- ANKI. *Anki – Powerful, intelligent flashcards*. 2026. Acesso em: mar. 2026. Disponível em: <<https://apps.ankiweb.net>>.
- Basecamp. *Hotwire: HTML over the wire*. 2025. Acesso em: mar. 2026. Disponível em: <<https://hotwired.dev>>.
- BJORK, R. A. Memory and metamemory considerations in the training of human beings. In: METCALFE, J.; SHIMAMURA, A. (Ed.). *Metacognition: Knowing about Knowing*. Cambridge: MIT Press, 1994. p. 185–205.
- CEPEDA, N. J. et al. Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, v. 132, n. 3, p. 354–380, maio 2006.
- Chart.js Contributors. *Chart.js: Simple yet flexible JavaScript charting library*. 2025. Acesso em: mar. 2026. Disponível em: <<https://www.chartjs.org>>.
- dry-rb Team. *dry-monads: Useful, common monads in idiomatic Ruby*. 2025. Acesso em: mar. 2026. Disponível em: <<https://dry-rb.org/gems/dry-monads>>.
- EBBINGHAUS, H. *Memória: uma contribuição à psicologia experimental*. New York: Teachers College, Columbia University, 1913. Tradução de Henry A. Ruger e Clara E. Bussenius. Original publicado em alemão: *Über das Gedächtnis*. Leipzig: Duncker & Humblot, 1885.
- GitHub. *ViewComponent: A framework for building reusable, testable & encapsulated view components in Ruby on Rails*. 2025. Acesso em: mar. 2026. Disponível em: <<https://viewcomponent.org>>.
- HANSSON, D. H. et al. *Ruby on Rails Guides*. 2026. Acesso em: mar. 2026. Disponível em: <<https://guides.rubyonrails.org>>.
- Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP). *Censo da Educação Superior 2022: notas estatísticas*. Brasília: INEP, 2023. Acesso em: mar. 2026. Disponível em: <<https://www.gov.br/inep/pt-br/areas-de-atuacao/pesquisas-estatisticas-e-indicadores/censo-da-educacao-superior>>.
- KARPICKE, J. D.; ROEDIGER, H. L. The critical importance of retrieval for learning. *Science*, v. 319, n. 5865, p. 966–968, fev 2008.
- KASNECI, E. et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*, v. 103, p. 102274, abr 2023.
- KRUGER, J.; DUNNING, D. Unskilled and unaware of it: How difficulties in recognizing one's own incompetence lead to inflated self-assessments. *Journal of Personality and Social Psychology*, v. 77, n. 6, p. 1121–1134, dez 1999.
- OpenAI. *OpenAI API Reference – Chat Completions*. 2026. Acesso em: mar. 2026. Disponível em: <<https://platform.openai.com/docs/api-reference/chat>>.

PERHAM, M. *Sidekiq: Simple, efficient background processing for Ruby*. 2025. Acesso em: mar. 2026. Disponível em: <<https://sidekiq.org>>.

Quizlet Inc. *Quizlet: Study Tools & Learning Resources*. 2026. Acesso em: mar. 2026. Disponível em: <<https://quizlet.com>>.

RemNote Inc. *RemNote – the all-in-one tool for thinking and learning*. 2026. Acesso em: mar. 2026. Disponível em: <<https://www.remnote.com>>.

ROEDIGER, H. L.; KARPICKE, J. D. Test-enhanced learning: Taking memory tests improves long-term retention. *Psychological Science*, v. 17, n. 3, p. 249–255, mar 2006.

SuperMemo World. *SuperMemo – the world's most efficient learning software*. 2026. Acesso em: mar. 2026. Disponível em: <<https://www.supermemo.com>>.

Tailwind Labs. *Tailwind CSS: A utility-first CSS framework*. 2025. Acesso em: mar. 2026. Disponível em: <<https://tailwindcss.com>>.

The PostgreSQL Global Development Group. *PostgreSQL: The World's Most Advanced Open Source Relational Database*. 2025. Acesso em: mar. 2026. Disponível em: <<https://www.postgresql.org>>.

WOZNIAK, P. A. *Algorithm SM-2*. 1987. SuperMemo Help. Acesso em: mar. 2026. Disponível em: <<https://super-memory.com/english/ol/sm2.htm>>.

WOZNIAK, P. A.; GORZELAŃCZYK, E. J. Optimization of repetition spacing in the practice of learning. *Acta Neurobiologiae Experimentalis*, v. 54, n. 1, p. 59–62, 1994.

6 DECLARAÇÃO DO AUTOR SOBRE A ENTREGA DO CÓDIGO-FONTE

Declaro, para os devidos fins, que o código-fonte integral do sistema, incluindo a estrutura de diretórios, as configurações de ambiente e os arquivos necessários à execução e à compilação do projeto, foi entregue à coordenação do curso como requisito para a conclusão do trabalho até a data de sua apresentação. O código-fonte completo permanece disponível para consulta e auditoria no repositório público: <<https://github.com/nelsonwenner/bigmemo>>.

7 DECLARAÇÃO DA COORDENAÇÃO DO CURSO SOBRE A ENTREGA DO CÓDIGO-FONTE



Ministério da Educação
Instituto Federal de Educação, Ciência e Tecnologia do Piauí
IFPI - CAMPUS TERESINA CENTRAL
Praça da Liberdade, 1597, Centro, TERESINA / PI, CEP 64.000-040
Fone: Site: www.ifpi.edu.br

DECLARAÇÃO 16/2026 - CCADS/DIAS/DENS/DG-TERCENT/CATCE/IFPI

TERESINA, 4 de junho de 2026.

Declaro, para os devidos fins, que o aluno do curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Nelson Dias Medeiros Neto, matrícula 20181ADS0027, entregou à coordenação do referido curso o código-fonte do relatório de software (Trabalho de Conclusão de Curso).

ALINE MONTENEGRO LEAL

Coordenadora do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

Documento assinado eletronicamente por:

■ **Aline Montenegro Leal**, COORDENADOR(A) DE CURSO - FUC0001 - CCADS-IFPI - CAMPUS TERESINA CENTRAL, em 04/06/2026 19:39:26.

Este documento foi emitido pelo SUAP em 04/06/2026. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpi.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código Verificador: 457058
Código de Autenticação: f8d61d02ab

