RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

Arquitetura multi-tenant para plataformas SaaS de educação a distância: escalabilidade, isolamento de dados e eficiência de custos

Multi-tenant architecture for distance-learning SaaS platforms: scalability, data isolation and cost efficiency

Arquitectura multi-tenant para plataformas SaaS de educación a distancia: escalabilidad, aislamiento de datos y eficiencia de costos

Gabriel Moraes
Tecnólogo em Análise e Desenvolvimento de Sistemas (Universidade Estácio de Sá). MBA em Big Data e Inteligência Competitiva e Pós-graduação em Desenvolvimento Full Stack (Descomplica). CEO e fundador da Oryn Labs, Orlando, Flórida, Estados Unidos.
E-mail: gabrielmoraesp@gmail.com

RESUMO
A educação a distância tornou-se a principal porta de entrada no ensino superior brasileiro: em 2021, 62,8% dos ingressantes optaram por cursos a distância. Esse crescimento pressiona as plataformas de ensino a atender um número elevado de instituições, cursos e alunos com custos controlados, o que favorece o modelo de software como serviço (SaaS) com arquitetura multi-tenant, em que uma única instância da aplicação atende a vários clientes. Este artigo discute, com base na literatura publicada até 2022, as decisões arquiteturais envolvidas nesse modelo: os níveis de maturidade de aplicações SaaS, os modelos de isolamento de dados (silo, ponte e pool), as técnicas de mapeamento de esquema em bancos de dados multi-tenant, a resolução do tenant, a escalabilidade horizontal, a observabilidade e a alocação de custos por cliente, bem como as exigências da Lei Geral de Proteção de Dados. Propõe-se uma arquitetura de referência em camadas e uma matriz de decisão para a escolha do modelo de isolamento por segmento de cliente. Conclui-se que a arquitetura híbrida, com pool para a longa cauda de pequenas instituições e silo ou ponte para clientes com exigências contratuais, oferece o melhor equilíbrio entre isolamento, eficiência e custo.
Palavras-chave: SaaS; multi-tenancy; educação a distância; arquitetura de software; computação em nuvem.

ABSTRACT
Distance learning became the main entry route into Brazilian higher education, with 62.8% of new students in 2021. This growth favours software-as-a-service platforms with multi-tenant architecture. Based on the literature published up to 2022, this paper discusses SaaS maturity levels, data-isolation models (silo, bridge and pool), multi-tenant schema-mapping techniques, tenant resolution, horizontal scalability, observability, per-tenant cost allocation and data-protection requirements, and proposes a layered reference architecture and a decision matrix for isolation models.
Keywords: SaaS; multi-tenancy; distance learning; software architecture; cloud computing.

1 INTRODUÇÃO

A educação a distância (EAD) passou por transformação acelerada na última década. Segundo o Censo da Educação Superior, a participação da EAD entre os ingressantes do ensino superior brasileiro passou de 18,4% em 2011 para 62,8% em 2021; no mesmo período, o número de ingressantes em cursos a distância cresceu 474%, enquanto o de ingressantes em cursos presenciais caiu 23,4% (INEP, 2022). Em 2021, as matrículas a distância já representavam 41,4% do total (Figura

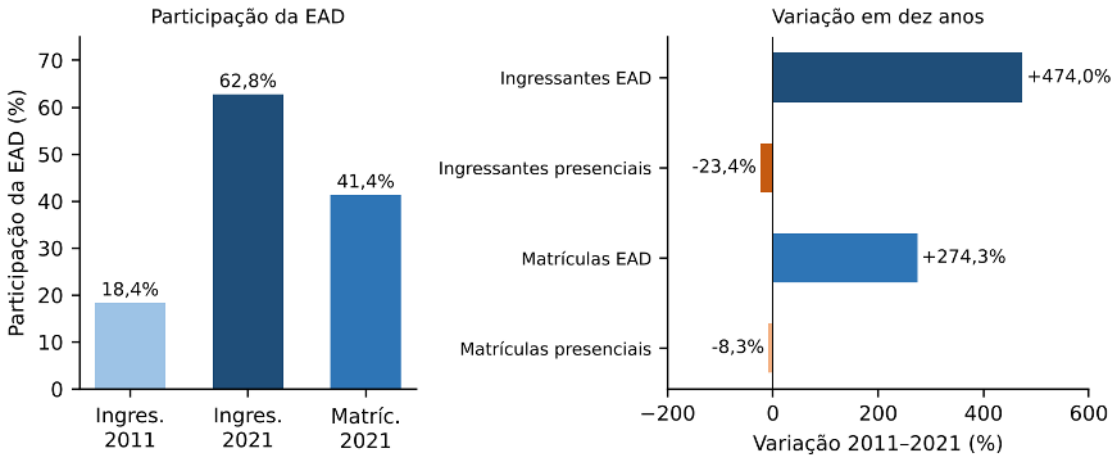
Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS - Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

1).

Figura 1 – Participação e variação da educação a distância no ensino superior brasileiro, 2011–2021



Fonte: elaborado pelo autor com dados do INEP, Censo da Educação Superior 2021 (INEP, 2022).

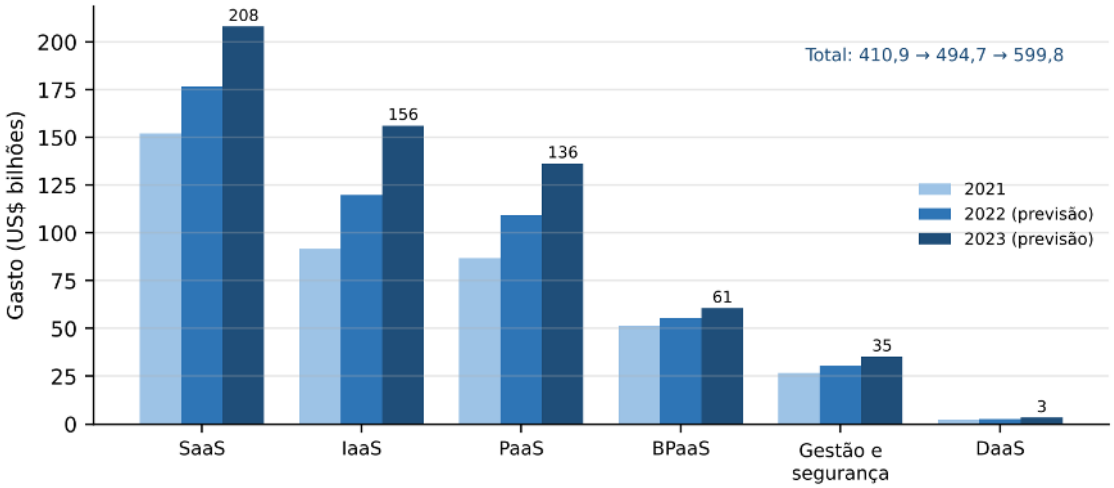
Esse movimento ampliou a demanda por plataformas de ensino capazes de atender muitas instituições simultaneamente, de faculdades de grande porte a escolas de cursos livres e preparatórios. Para essas plataformas, o modelo de software como serviço (SaaS) é economicamente atraente: a aplicação é operada pelo fornecedor e acessada pela internet, e o custo de infraestrutura é diluído entre os clientes (ARMBRUST et al., 2010; MELL; GRANCE, 2011). O mercado acompanha essa tendência. A Gartner estimou que o gasto mundial com serviços de nuvem pública atingiria US$ 494,7 bilhões em 2022, dos quais US$ 176,6 bilhões em aplicações SaaS, o maior segmento (Figura 2) (GARTNER, 2022).

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

Figura 2 – Gasto mundial com serviços de nuvem pública por segmento, 2021–2023



Fonte: elaborado pelo autor com dados da Gartner (2022). Valores de 2022 e 2023 são previsões.

A viabilidade do SaaS, porém, depende de uma decisão arquitetural central: como compartilhar recursos entre clientes, denominados *tenants*, sem comprometer o isolamento dos dados, o desempenho e a capacidade de personalização. O objetivo deste artigo é discutir essa decisão para plataformas de EAD, organizando os conceitos, as alternativas e os critérios de escolha disponíveis na literatura técnica publicada até 2022.

2 METODOLOGIA

Trata-se de revisão narrativa, de caráter aplicado, da literatura sobre arquitetura de aplicações SaaS multi-tenant. Foram consultados artigos de conferências e periódicos sobre multi-tenancy (BEZEMER; ZAIDMAN, 2010; KREBS; MOMM; KOUNEV, 2012; AULBACH et al., 2008), obras de referência em arquitetura de software e sistemas de dados (BASS; CLEMENTS; KAZMAN, 2021; KLEPPMANN, 2017; NEWMAN, 2021), documentos técnicos de provedores e organismos de normalização (MELL; GRANCE, 2011; CHONG; CARRARO, 2006; AWS, 2021; IMS GLOBAL LEARNING CONSORTIUM, 2019) e dados oficiais sobre a EAD no Brasil (INEP, 2022). Os resultados foram organizados em fundamentos, modelos de isolamento, camada de dados, arquitetura de referência, escalabilidade e custo, e requisitos de proteção de dados.

3 FUNDAMENTOS DO SAAS MULTI-TENANT

O NIST define a computação em nuvem a partir de cinco características essenciais: autosserviço sob demanda, amplo acesso pela rede, agrupamento de recursos, elasticidade rápida e serviço medido; o

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

SaaS é o modelo de serviço em que o consumidor utiliza aplicações do provedor executadas em infraestrutura de nuvem (MELL; GRANCE, 2011). Armbrust et al. (2010) destacam que a economia da nuvem decorre da elasticidade e do pagamento por uso, que transferem ao provedor o risco de dimensionamento.

A multi-tenancy é a propriedade de uma aplicação que atende a vários clientes a partir de uma instância compartilhada, mantendo para cada um a percepção de um sistema dedicado (BEZEMER; ZAIDMAN, 2010; KREBS; MOMM; KOUNEV, 2012). Chong e Carraro (2006) propuseram um modelo de maturidade em quatro níveis: no primeiro, cada cliente tem sua própria versão personalizada da aplicação; no segundo, todos usam o mesmo código, configurado por metadados, mas em instâncias separadas; no terceiro, uma única instância atende a todos os clientes; no quarto, essa instância é replicada atrás de um balanceador de carga, permitindo escala horizontal. A Tabela 1 resume esses níveis.

Tabela 1 – Níveis de maturidade de aplicações SaaS

Nível	Característica	Implicação para plataformas de EAD
1. Ad hoc	Instância e código personalizados por cliente	Custo de manutenção cresce com cada instituição
2. Configurável	Mesmo código, configurado por metadados; instâncias separadas	Atualizações uniformes, mas infraestrutura duplicada
3. Multi-tenant	Instância única atende a todos os clientes	Ganho de eficiência; exige isolamento lógico rigoroso
4. Escalável	Instância multi-tenant replicada com balanceamento	Escala horizontal para picos de matrícula e provas

Fonte: elaborado pelo autor com base em Chong e Carraro (2006).

Bezemer e Zaidman (2010) observam que a multi-tenancy reduz custos de infraestrutura e de manutenção, pois há uma única base de código e uma única implantação a atualizar, mas aumenta a complexidade do software, que passa a precisar de mecanismos explícitos de isolamento, configuração por cliente e controle de desempenho. Krebs, Momm e Kounev (2012) identificam como preocupações arquiteturais centrais o isolamento de dados, o isolamento de desempenho, a personalização, a escalabilidade e a segurança.

4 MODELOS DE ISOLAMENTO

A literatura técnica e a documentação de provedores de nuvem consolidaram três modelos básicos de

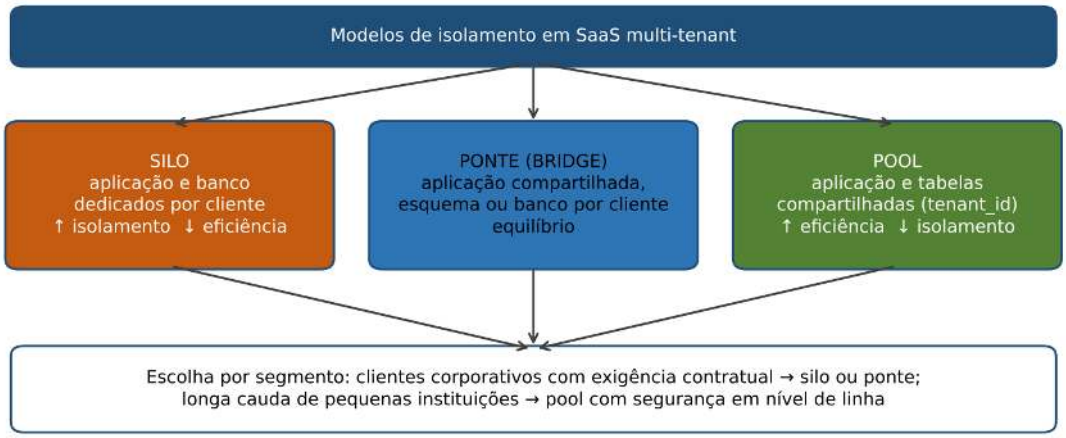
Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

isolamento (AWS, 2021). No modelo *silo*, cada cliente dispõe de infraestrutura dedicada, inclusive banco de dados; no modelo *pool*, todos compartilham aplicação e tabelas, e os registros são diferenciados por um identificador de tenant; no modelo *ponte* (*bridge*), a aplicação é compartilhada, mas cada cliente tem esquema ou banco próprio (Figura 3).

Figura 3 – Modelos de isolamento em aplicações SaaS multi-tenant



Fonte: elaborado pelo autor com base em AWS (2021), Krebs, Momm e Kounev (2012) e Bezemer e Zaidman (2010).

A escolha envolve compromissos claros. O silo oferece o maior isolamento e facilita atender exigências contratuais de grandes instituições, como residência de dados, janelas próprias de manutenção e restauração individual de backups, mas multiplica custos e esforço operacional. O pool maximiza a eficiência e simplifica a operação, mas exige que o isolamento seja garantido em todas as consultas, o que torna a aplicação mais sensível a falhas de programação, e expõe os clientes ao problema do "vizinho barulhento", em que um tenant com carga elevada degrada o desempenho dos demais (KREBS; MOMM; KOUNEV, 2012). A ponte é um meio-termo que preserva parte do isolamento com custo intermediário.

Tabela 2 – Matriz de decisão para o modelo de isolamento por segmento de cliente

Critério	Silo	Ponte	Pool
Isolamento de dados	Físico	Lógico forte (esquema/banco)	Lógico por linha
Custo por cliente	Alto	Médio	Baixo
Esforço operacional	Alto	Médio	Baixo

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS – Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

Critério	Silo	Ponte	Pool
Restauração individual	Simples	Moderada	Complexa
Risco de vizinho barulhento	Baixo	Médio	Alto
Segmento indicado	Grandes instituições com exigência contratual	Instituições médias	Longa cauda de pequenas escolas

Fonte: elaborado pelo autor com base em AWS (2021) e Krebs, Momm e Kounev (2012).

Em plataformas de EAD, a base de clientes costuma ser heterogênea: poucas instituições de grande porte, com milhares de alunos e exigências contratuais, e muitas escolas pequenas, com centenas de alunos e sensibilidade a preço. Por isso, a arquitetura híbrida, que combina pool para a longa cauda e silo ou ponte para os clientes de maior porte, tende a ser a mais adequada (Tabela 2). O requisito técnico para que esse modelo funcione é que o código da aplicação seja o mesmo em todos os casos, variando apenas a camada de roteamento para os dados.

5 A CAMADA DE DADOS

Aulbach et al. (2008) analisaram as técnicas de mapeamento de esquema para bancos de dados multi-tenant: tabelas privadas por cliente, tabelas de extensão, tabela universal, tabelas pivô e tabelas em blocos (*chunk tables*), estas últimas propostas pelos autores para conciliar flexibilidade de personalização e desempenho. Na prática atual, a abordagem mais difundida para o modelo pool é a coluna de identificação do tenant em todas as tabelas, combinada com índices compostos e com políticas de segurança em nível de linha, recurso disponível no PostgreSQL desde a versão 9.5 (POSTGRESQL GLOBAL DEVELOPMENT GROUP, 2021). Com essas políticas, o próprio banco de dados filtra as linhas conforme o tenant da sessão, reduzindo o risco de que um erro na aplicação exponha dados de outro cliente.

Para os campos personalizados que cada instituição deseja acrescentar, como dados específicos de matrícula ou de avaliação, as alternativas incluem colunas semiestruturadas (por exemplo, JSON) ou tabelas de extensão (AULBACH et al., 2008; KLEPPMANN, 2017). Os arquivos de maior volume, como vídeos de aulas e materiais, devem ser armazenados em serviço de objetos, com prefixos por tenant e controle de acesso por URLs assinadas, enquanto os dados analíticos são replicados para um data warehouse (KLEPPMANN, 2017).

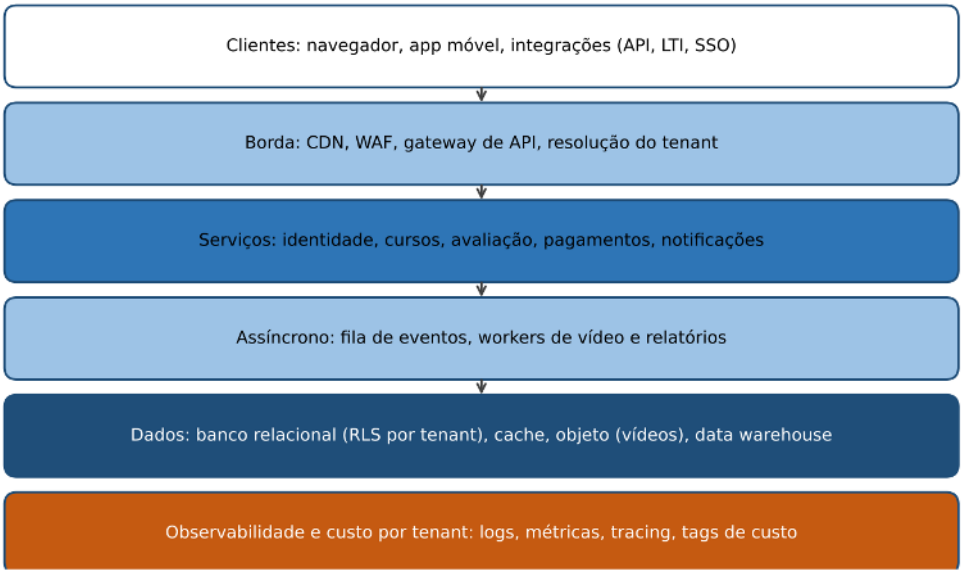
Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

RCMOS - Revista Científica Multidisciplinar O Saber.
ISSN: 2675-9128. São Paulo-SP.
Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

6 ARQUITETURA DE REFERÊNCIA

A Figura 4 apresenta uma arquitetura de referência em camadas para plataformas de EAD multi-tenant, elaborada a partir dos princípios discutidos.

Figura 4 – Arquitetura de referência em camadas para plataforma de EAD multi-tenant



Fonte: elaborado pelo autor com base em Bass, Clements e Kazman (2021), Kleppmann (2017) e AWS (2021).

Na camada de borda, a resolução do tenant identifica a instituição a partir do domínio, do subdomínio ou do token de autenticação, e propaga essa identificação para todas as camadas seguintes. A camada de serviços concentra os domínios funcionais da plataforma, como identidade, cursos, avaliação, pagamentos e notificações; a integração com ambientes externos de aprendizagem pode seguir a especificação LTI 1.3 (IMS GLOBAL LEARNING CONSORTIUM, 2019). Tarefas pesadas, como transcodificação de vídeos, geração de certificados e relatórios, são executadas de forma assíncrona por filas e *workers*, o que protege a experiência do aluno nos períodos de pico (KLEPPMANN, 2017; NEWMAN, 2021). A camada de observabilidade registra logs, métricas e rastreamento com o identificador do tenant, permitindo diagnosticar problemas por cliente e atribuir custos (BEYER et al., 2016).

7 ESCALABILIDADE E CUSTO POR TENANT

Plataformas de EAD apresentam padrões de carga sazonais e concentrados: picos no início dos

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

semestres, nos prazos de entrega de atividades e nos períodos de provas. A escala horizontal, com instâncias sem estado atrás de balanceadores de carga e ajuste automático de capacidade, é a estratégia indicada para absorver esses picos (ARMBRUST et al., 2010; KLEPPMANN, 2017). O banco de dados é geralmente o gargalo; as técnicas usuais incluem réplicas de leitura, cache de consultas frequentes e, em escala maior, particionamento por tenant (KLEPPMANN, 2017).

O custo por cliente é um indicador econômico essencial no SaaS. Em arquiteturas pool, ele não é observado diretamente, pois os recursos são compartilhados, e precisa ser estimado a partir de métricas de uso, como requisições, armazenamento, minutos de vídeo e tempo de processamento, associadas ao identificador do tenant (AWS, 2021). Essa estimativa permite verificar se os planos de preço cobrem o custo de atendimento de cada segmento e identificar clientes cujo perfil de uso justifica migração para o modelo ponte ou silo.

8 PROTEÇÃO DE DADOS

Plataformas de EAD tratam dados pessoais de alunos, como identificação, desempenho acadêmico e, frequentemente, dados de menores de idade. A Lei Geral de Proteção de Dados (Lei nº 13.709/2018) estabelece princípios como finalidade, necessidade, segurança e prevenção, e exige medidas técnicas e administrativas aptas a proteger os dados de acessos não autorizados (BRASIL, 2018). Na relação típica do SaaS educacional, a instituição de ensino atua como controladora e o fornecedor da plataforma como operador, o que deve ser refletido em contrato. Do ponto de vista arquitetural, isso se traduz em isolamento verificável entre tenants, criptografia em trânsito e em repouso, registros de acesso, procedimentos de exclusão e portabilidade de dados por cliente e testes periódicos de segurança (BRASIL, 2018; KREBS; MOMM; KOUNEV, 2012).

9 DISCUSSÃO

A literatura revisada mostra que não existe um modelo de isolamento universalmente superior. O modelo pool é o que torna o SaaS economicamente viável para a longa cauda de pequenas instituições, mas transfere para a aplicação e para o banco de dados a responsabilidade pelo isolamento (BEZEMER; ZAIDMAN, 2010; KREBS; MOMM; KOUNEV, 2012). O silo é a resposta adequada para clientes de grande porte com exigências específicas, mas não escala como modelo único. A arquitetura híbrida, com o mesmo código e roteamento de dados por tenant, permite que a plataforma ofereça níveis de isolamento diferentes como parte da estratégia comercial (AWS, 2021).

Duas condições técnicas são decisivas para o sucesso. A primeira é tratar o identificador do tenant

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

como elemento de primeira classe em todas as camadas, do roteamento à observabilidade. A segunda é automatizar o provisionamento e a migração de tenants entre modelos, de modo que um cliente possa ser movido do pool para um banco dedicado sem alteração de código (NEWMAN, 2021; BASS; CLEMENTS; KAZMAN, 2021). Sem essas condições, a plataforma tende a regredir para instâncias personalizadas, o primeiro nível de maturidade de Chong e Carraro (2006), com perda das vantagens do modelo SaaS.

10 CONSIDERAÇÕES FINAIS

O crescimento da educação a distância no Brasil, que em 2021 já respondia pela maioria dos ingressantes no ensino superior (INEP, 2022), cria demanda por plataformas que atendam muitas instituições com qualidade e custo controlado. A arquitetura multi-tenant é a base técnica dessa escala. Este artigo organizou as principais decisões envolvidas, como o nível de maturidade, o modelo de isolamento, a camada de dados, a arquitetura em camadas, a escalabilidade, o custo por tenant e a proteção de dados, e propôs uma matriz de decisão por segmento de cliente. Como desdobramento, recomenda-se a avaliação empírica do custo por tenant e do isolamento de desempenho em plataformas em operação.

REFERÊNCIAS

AMAZON WEB SERVICES (AWS). **SaaS Lens**: AWS Well-Architected Framework. Seattle: Amazon Web Services, 2021.

ARMBRUST, M. *et al*. A view of cloud computing. **Communications of the ACM**, v. 53, n. 4, p. 50–58, 2010.

AULBACH, S. *et al*. Multi-tenant databases for software as a service: schema-mapping techniques. *In*: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 2008, Vancouver. **Proceedings** [...]. New York: ACM, 2008. p. 1195–1206.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software architecture in practice**. 4. ed. Boston: Addison-Wesley, 2021.

BEYER, B. *et al*. (ed.). **Site reliability engineering**: how Google runs production systems. Sebastopol: O'Reilly, 2016.

BEZEMER, C.-P.; ZAIDMAN, A. Multi-tenant SaaS applications: maintenance dream or nightmare? *In*: JOINT ERCIM WORKSHOP ON SOFTWARE EVOLUTION (EVOL) AND INTERNATIONAL WORKSHOP ON PRINCIPLES OF SOFTWARE EVOLUTION (IWPSE), 2010, Antwerp. **Proceedings** [...]. New York: ACM, 2010.

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.

Ano VII, v.2 2026 | submissão: 02/09/2026 | aceito: 07/10/2026 | publicação: 12/11/2026

BRASIL. Lei nº 13.709, de 14 de agosto de 2018. Lei Geral de Proteção de Dados Pessoais (LGPD). **Diário Oficial da União**, Brasília, DF, 15 ago. 2018.

CHONG, F.; CARRARO, G. **Architecture strategies for catching the long tail**. Redmond: Microsoft Corporation, 2006.

GARTNER. **Gartner forecasts worldwide public cloud end-user spending to reach nearly $500 billion in 2022**. Stamford: Gartner, 19 abr. 2022. Comunicado de imprensa.

IMS GLOBAL LEARNING CONSORTIUM. **Learning Tools Interoperability core specification**: version 1.3. Lake Mary: IMS Global, 2019.

INSTITUTO NACIONAL DE ESTUDOS E PESQUISAS EDUCACIONAIS ANÍSIO TEIXEIRA (INEP). **Censo da Educação Superior 2021**: notas estatísticas. Brasília, DF: INEP, 2022.

KLEPPMANN, M. **Designing data-intensive applications**. Sebastopol: O'Reilly, 2017.

KREBS, R.; MOMM, C.; KOUNEV, S. Architectural concerns in multi-tenant SaaS applications. *In*: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING AND SERVICES SCIENCE (CLOSER), 2., 2012, Porto. **Proceedings** [...]. Setúbal: SciTePress, 2012.

MELL, P.; GRANCE, T. **The NIST definition of cloud computing**. Gaithersburg: National Institute of Standards and Technology, 2011. (Special Publication 800-145).

NEWMAN, S. **Building microservices**: designing fine-grained systems. 2. ed. Sebastopol: O'Reilly, 2021.

POSTGRESQL GLOBAL DEVELOPMENT GROUP. **PostgreSQL 14 documentation**: row security policies. 2021. Disponível em: https://www.postgresql.org/docs/14/ddl-rowsecurity.html.

Este é um artigo publicado em acesso aberto (Open Access) sob a licença CreativeCommons Attribution, que permite uso, distribuição e reprodução em qualquer meio, sem restrições desde que o trabalho original seja corretamente citado.