REVISTA TÓPICOS

ARQUITETURA CELULAR NA COMPUTAÇÃO EM NUVEM: UMA REVISÃO SISTEMÁTICA SOBRE RESILIÊNCIA, ESCALABILIDADE E TENDÊNCIAS EMERGENTES

DOI: 10.5281/zenodo.16474011

Cláudio Filipe Lima Rapôso¹

RESUMO

Este artigo apresenta uma revisão sistemática da literatura sobre a aplicação da Cell-Based Architecture (CBA) na computação em nuvem, com foco nos impactos sobre a resiliência, escalabilidade e governança de sistemas distribuídos. A análise de 28 documentos selecionados entre 2022 e 2025 revelou que a CBA promove isolamento de falhas, escalabilidade seletiva e redução do tempo de recuperação de incidentes, por meio da segmentação lógica de aplicações em células autônomas. Estudos de caso em empresas como Slack e DoorDash demonstraram benefícios concretos, mas também destacaram desafios relacionados à complexidade operacional, replicação de infraestrutura e consistência de dados. A discussão abordou ainda tendências emergentes, como o uso de algoritmos de detecção de comunidades, integração com service mesh e inteligência artificial para autoescalamento. Conclui-se que a CBA representa um modelo arquitetônico promissor, cuja adoção requer maturidade técnica, automação

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

avançada e alinhamento organizacional com práticas modernas de engenharia de software.

Palavras-chave: Cell-Based Architecture, Computação em Nuvem, Resiliência, Escalabilidade, Sistemas Distribuídos.

ABSTRACT

This article presents a systematic literature review on the application of Cell-Based Architecture (CBA) in cloud computing, focusing on its impacts on resilience, scalability, and governance of distributed systems. The analysis of 28 documents published between 2022 and 2025 revealed that CBA enables fault isolation, selective scalability, and reduced incident recovery times by segmenting applications into autonomous logical cells. Case studies from companies such as Slack and DoorDash demonstrated tangible benefits but also highlighted challenges regarding operational complexity, infrastructure replication, and data consistency. The discussion also covered emerging trends such as the use of community detection algorithms, integration with service mesh, and artificial intelligence for autonomous scaling. It is concluded that CBA represents a promising architectural model, whose adoption requires technical maturity, advanced automation, and organizational alignment with modern software engineering practices.

Keywords: Cell-Based Architecture, Cloud Computing, Resilience, Scalability, Distributed Systems.

1 INTRODUÇÃO

A transformação digital tem provocado mudanças profundas na maneira como sistemas computacionais são projetados, desenvolvidos e operados.

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

Em um cenário cada vez mais orientado por escalabilidade, disponibilidade contínua e tolerância a falhas, torna-se evidente a necessidade de abordagens arquitetônicas que respondam de forma eficiente às exigências de desempenho e confiabilidade impostas por aplicações modernas. Nesse contexto, a computação em nuvem, ou Cloud Computing, desponta como um paradigma consolidado, permitindo o provisionamento sob demanda de recursos computacionais e oferecendo elasticidade e resiliência como propriedades fundamentais (Mell & Grance, 2011).

Com o avanço do modelo de microsserviços, onde aplicações são decompostas em serviços menores, independentes e comunicantes, emergem desafios associados ao gerenciamento de falhas, consistência de dados e escalabilidade granular. O padrão arquitetural de microsserviços, embora traga benefícios claros em termos de modularidade e facilidade de manutenção, também introduz complexidades operacionais significativas, como o aumento do número de interações entre componentes, a propagação de falhas e a dificuldade de orquestração em larga escala (Fowler & Lewis, 2014). Nesse contexto, torna-se evidente a necessidade de estratégias que proporcionem isolamento eficaz e governança mais refinada sobre os domínios de aplicação.

É nesse ponto que a Cell-Based Architecture (CBA) se apresenta como uma proposta inovadora e robusta. Inspirada no padrão bulkhead, utilizado originalmente na engenharia naval para compartimentalização de compartimentos a fim de evitar que uma falha local comprometa a totalidade da embarcação, a CBA aplica este princípio ao desenvolvimento de sistemas

REVISTA TÓPICOS

distribuídos. Cada célula representa uma instância autônoma e autocontida de uma parte do sistema, composta por microserviços, banco de dados, cache, lógica de negócio e mecanismos de observabilidade, todos isolados das demais células (Pisani & Gancarz, 2024). Esse isolamento estrutural permite que falhas ocorridas em uma célula não afetem as demais, mitigando o chamado blast radius e promovendo maior resiliência sistêmica.

A arquitetura celular tem ganhado notoriedade principalmente no contexto da computação em nuvem, onde escalabilidade horizontal, replicação geográfica e separação de domínios são características desejáveis. Empresas como Amazon, Slack, Netflix e DoorDash passaram a empregar modelos celulares em seus sistemas de produção, buscando otimizar o gerenciamento de carga, reduzir o tempo médio de recuperação (Mean Time To Recovery - MTTR) e maximizar o tempo médio entre falhas (Mean Time Between Failures - MTBF), conforme documentado em suas apresentações e white papers técnicos (AWS, 2024; Rackspace, 2024).

Do ponto de vista conceitual, a Cell-Based Architecture fundamenta-se na divisão do sistema em múltiplas células, cada uma operando de forma independente, porém coordenada por um cell router. Esse roteador de célula é responsável por encaminhar requisições aos respectivos domínios com base em uma chave de partição, como o ID do cliente ou do recurso requisitado. O controle centralizado é evitado, dando lugar a uma lógica de distribuição que respeita a autonomia das unidades celulares, permitindo sua atualização, replicação ou desativação sem impacto nos demais componentes do sistema (AWS, 2023). Além disso, a camada de controle, ou control

REVISTA TÓPICOS

plane, é responsável pela orquestração das células, gerenciando sua criação, configuração e monitoramento, garantindo observabilidade e integridade arquitetural.

A adoção de CBA em ambientes de nuvem é favorecida pela existência de recursos como zonas de disponibilidade, redes virtuais privadas (VPCs), contas segregadas e orquestradores como Kubernetes. Tais ferramentas proporcionam o isolamento físico e lógico necessário à implementação de células autônomas, além de viabilizar a governança sobre políticas de segurança, tráfego e escalonamento (Tharmarajah, 2024). O paradigma se mostra particularmente eficiente em aplicações que lidam com grande número de usuários ou requisições simultâneas, em que o particionamento do tráfego e o tratamento diferenciado por domínios favorecem o desempenho e a estabilidade da aplicação como um todo.

Contudo, apesar de suas vantagens evidentes, a arquitetura baseada em células também apresenta desafios relevantes. A complexidade de gerenciamento das células, a sobrecarga operacional introduzida pelo aumento no número de instâncias e serviços, a necessidade de consistência de dados entre células e a dependência de mecanismos sofisticados de roteamento e monitoramento configuram pontos críticos que precisam ser cuidadosamente avaliados (Re:Invent, 2024). Além disso, a própria estrutura do cell router pode representar um ponto de falha, exigindo alta disponibilidade e resiliência própria, sob pena de comprometer a eficácia de todo o modelo arquitetônico (Rackspace, 2024).

REVISTA TÓPICOS

A crescente adoção da CBA por grandes empresas e plataformas críticas, no entanto, indica uma tendência consolidada rumo à modularização radical de sistemas em ambientes distribuídos. Estudos recentes demonstram que sua aplicação em clusters Kubernetes, por exemplo, permite particionamento por domínios de negócios, como pagamentos, usuários e catálogo, garantindo isolamento completo e tolerância a falhas específicas sem perda de funcionalidade geral (Mohanagandhi & Ramalingam, 2024). Essa abordagem, além de facilitar o escalonamento seletivo de partes do sistema, também promove o desenvolvimento independente de equipes, alinhando-se aos princípios da engenharia organizacional moderna e à adoção de práticas DevOps.

A presente pesquisa se justifica pela necessidade de sistematização do conhecimento acumulado em torno da arquitetura celular aplicada à computação em nuvem, dada sua crescente relevância e potencial de impacto na engenharia de software contemporânea. Embora diversas fontes técnicas e relatos de experiências práticas estejam disponíveis, há carência de estudos acadêmicos que analisem criticamente a CBA sob o ponto de vista metodológico, identificando seus benefícios e limitações, bem como suas implicações para a evolução da arquitetura de sistemas distribuídos.

Assim, esta revisão bibliográfica tem por objetivo central analisar a Cell-Based Architecture como estratégia arquitetônica para ambientes de Cloud Computing, explorando seus fundamentos, aplicações, vantagens, limitações e lacunas de pesquisa. Pretende-se, com isso, contribuir para o aprofundamento teórico e prático sobre o tema, fornecendo uma base sólida

REVISTA TÓPICOS

para profissionais da área e pesquisadores interessados em arquiteturas resilientes e escaláveis.

As buscas foram realizadas entre os meses de maio e julho de 2025, utilizando-se as seguintes bases de dados acadêmicas: IEEE Xplore, SpringerLink, Scopus e Google Scholar. Complementarmente, foram consultados repositórios técnicos especializados, como InfoQ, DZone, AWS Documentation e white papers de empresas como Rackspace e Netflix, devido à relevância prática da CBA no contexto corporativo. As palavras-chave utilizadas incluíram: "Cell-Based Architecture", "Cloud Computing", "Fault Isolation", "Resilient Systems", "Bulkhead Pattern", "Microservices Scalability", além de suas correspondentes em português. Empregaram-se operadores booleanos e filtros por data (2022–2025) para garantir atualidade e pertinência dos resultados.

Foram definidos os seguintes critérios de inclusão: artigos científicos revisados por pares, publicações técnicas reconhecidas, estudos de caso documentados e guias oficiais com fundamentação técnica. Os critérios de exclusão abrangeram documentos com ausência de fundamentação conceitual, duplicidades, publicações com foco em contextos muito específicos sem relação com sistemas distribuídos, bem como materiais sem vínculo com nuvem pública ou ambientes escaláveis. A triagem inicial resultou em 84 documentos, dos quais 28 foram selecionados após leitura integral e avaliação qualitativa. Esse corpus final incluiu 14 fontes internacionais e 14 nacionais ou de origem técnica especializada, garantindo diversidade epistemológica e representatividade geográfica.

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

A análise dos documentos seguiu abordagem qualitativa-descritiva, com ênfase na identificação de padrões conceituais, recorrência de temas, avaliação crítica de resultados e identificação de lacunas para pesquisas futuras. As informações extraídas foram organizadas em categorias analíticas pré-definidas: definição conceitual da CBA; estrutura arquitetônica; casos de uso; benefícios operacionais; desafios e limitações; implicações para o design de sistemas; e perspectivas de pesquisa. Os dados foram sistematizados com apoio de software Zotero, para organização das referências, e Mendeley, para anotações temáticas e categorização semântica.

Quanto à validade do estudo, adotou-se triangulação metodológica entre fontes teóricas (acadêmicas), empíricas (estudos de caso) e técnicas (documentações oficiais e white papers), visando mitigar viés interpretativo e assegurar consistência nos achados. Além disso, foram realizados processos de revisão cruzada entre os documentos, com checagem cruzada de argumentos e verificação de coerência nas evidências apresentadas. A originalidade do corpus foi testada com ferramentas de detecção de similaridade, assegurando que o conteúdo analisado não apresentasse sobreposição significativa com outras revisões publicadas.

A presente metodologia permite não apenas delinear o panorama atual da Cell-Based Architecture em nuvem, como também oferece subsídios metodológicos replicáveis para futuras investigações no campo da engenharia de software distribuída. Ao reunir dados atualizados, oriundos de fontes consolidadas e avaliados com critérios rigorosos, esta revisão

REVISTA TÓPICOS

bibliográfica sistemática contribui para a construção de uma base sólida de conhecimento sobre um tema emergente e de elevada relevância tecnológica.

Diante do exposto, é possível afirmar que a Cell-Based Architecture representa um marco evolutivo no design de sistemas em nuvem, refletindo uma busca constante por robustez, escalabilidade e resiliência. Sua compreensão, no entanto, exige não apenas o conhecimento das tecnologias envolvidas, mas também uma análise crítica de suas implicações operacionais e organizacionais. A partir deste estudo, busca-se lançar luz sobre essas dimensões e propor caminhos para sua adoção eficiente, consciente e tecnicamente fundamentada.

2 FUNDAMENTAÇÃO TEÓRICA E APLICAÇÕES DA CBA

A fundamentação teórica da Cell-Based Architecture (CBA) está ancorada nos princípios da engenharia de software resiliente e distribuída, alinhando-se a modelos arquitetônicos que visam modularidade, isolamento de falhas e escalabilidade autônoma. Desde a popularização do padrão arquitetural de microsserviços, arquitetos de sistemas têm buscado formas de mitigar o impacto das falhas sistêmicas que, mesmo em ambientes altamente distribuídos, continuam afetando o desempenho e a disponibilidade dos serviços. A CBA surge como uma resposta robusta a esse desafio, adotando a segmentação da aplicação em domínios independentes chamados "células", cada uma projetada para funcionar de maneira isolada, mas coordenada por um mecanismo de roteamento inteligente (Pisani & Gancarz, 2024).

REVISTA TÓPICOS

Do ponto de vista teórico, a CBA pode ser compreendida como uma extensão pragmática do padrão bulkhead, amplamente discutido em arquiteturas orientadas à resiliência. Em sua aplicação original, o bulkhead visava compartimentalizar navios para evitar que um dano localizado compromettesse toda a embarcação. Na computação distribuída, o princípio é adaptado para compartimentalizar componentes lógicos e físicos do sistema, criando barreiras estruturais que impedem a propagação de falhas entre domínios (Fowler & Lewis, 2014). A célula, nesse contexto, constitui um agrupamento coeso de serviços e dados, dotado de infraestrutura própria e políticas de segurança, escalonamento e monitoramento individualizadas (AWS, 2023).

Cada célula é projetada para operar de forma autônoma, sendo responsável por um subconjunto específico de usuários, recursos ou funções. Essa segmentação é facilitada pela introdução de uma chave de partição, geralmente baseada no ID do cliente ou na localização geográfica, que permite ao cell router encaminhar requisições à célula correspondente. Esse roteador funciona como um gateway central, mas não detém lógica de negócio, sendo responsável apenas pela determinação da célula-alvo com base em regras pré-configuradas. Essa separação clara entre lógica de roteamento e execução funcional fortalece a modularidade da arquitetura e permite implantações segmentadas por região, linha de produto ou tipo de serviço (AWS Well-Architected Framework, 2024).

Empresas como Slack e DoorDash têm adotado esse modelo para lidar com a complexidade de escalar serviços em tempo real. No caso do Slack, por

REVISTA TÓPICOS

exemplo, a migração para uma arquitetura celular permitiu a criação de instâncias isoladas por zona de disponibilidade (Availability Zone), com planos de controle independentes e mecanismos autônomos de recuperação de falhas. Essa abordagem garantiu maior continuidade de serviço, mesmo diante de interrupções regionais ou falhas de infraestrutura (DZone, 2024). Já no caso da DoorDash, a adoção do projeto SuperCell resultou em uma arquitetura em que cada célula representa uma réplica completa dos serviços críticos da aplicação, isolando falhas em usuários ou restaurantes específicos sem comprometer o funcionamento global da plataforma (Rackspace, 2024).

No contexto acadêmico, estudos recentes apontam para a aplicação da CBA em clusters Kubernetes, especialmente no Amazon EKS, onde a granularidade das células pode ser definida por domínios funcionais como catálogo de produtos, processamento de pagamentos ou autenticação de usuários. Essa implementação permite que cada célula seja implantada em sua própria namespace, com configurações de rede, política de segurança e métricas individualizadas. Além disso, o uso de orquestradores e operadores personalizados viabiliza a automação do ciclo de vida das células, reduzindo a carga operacional e promovendo maior consistência na aplicação de boas práticas de DevOps (Mohanagandhi & Ramalingam, 2024).

Os benefícios identificados na literatura sobre CBA são consistentes em diversos cenários. Em primeiro lugar, destaca-se o isolamento de falhas, um dos objetivos primários da arquitetura, que permite confinar problemas técnicos a uma célula específica, evitando a propagação para o restante do sistema. Em segundo lugar, a escalabilidade seletiva possibilita alocar

REVISTA TÓPICOS

recursos apenas para células com maior demanda, otimizando o uso de infraestrutura. Em terceiro lugar, a autonomia das células viabiliza a realização de testes canários, deploys incrementais e retomadas rápidas, reduzindo significativamente o tempo de recuperação após falhas (AWS, 2023; InfoQ, 2024).

Contudo, a adoção da CBA não é isenta de desafios. Um dos principais problemas apontados por desenvolvedores e arquitetos refere-se à complexidade operacional introduzida pela necessidade de manter múltiplas instâncias isoladas de serviços, bancos de dados e pipelines de integração contínua. A replicação de componentes e a duplicação de esforços de monitoramento, segurança e governança impõem custos elevados, que devem ser balanceados com os benefícios obtidos. Além disso, o cell router, apesar de sua simplicidade conceitual, torna-se um ponto crítico na arquitetura, exigindo alta disponibilidade, resiliência própria e mecanismos de fallback em caso de falha (AWS Re:Invent, 2024).

Outro ponto sensível refere-se à consistência de dados entre células. Em aplicações que exigem sincronia ou compartilhamento eventual de informações, torna-se necessário o uso de mecanismos como replicação assíncrona, event sourcing ou mensagens com garantias de entrega. Tais estratégias, embora viáveis, requerem cuidados especiais para evitar inconsistências lógicas e garantir integridade transacional em sistemas distribuídos. A literatura sugere que, em cenários de alta criticidade, a definição dos limites das células deve ser orientada por critérios de coesão funcional e independência de estado, o que reforça a necessidade de

REVISTA TÓPICOS

modelagem arquitetural precisa e baseada em domínio (Pisani & Gancarz, 2024; AWS, 2023).

As perspectivas de pesquisa em torno da CBA incluem a automatização do balanceamento de carga entre células, o desenvolvimento de métricas para avaliar a eficácia do isolamento e a otimização de estratégias de roteamento com base em aprendizado de máquina. Algumas abordagens recentes têm explorado o uso de algoritmos de detecção de comunidades, tradicionalmente aplicados em redes complexas, para identificar padrões de uso e agrupamentos naturais de serviços, permitindo uma divisão mais eficiente e adaptativa do sistema em células (Silva, J. R., & Almeida, F. T., 2025). Tais inovações sugerem que a CBA pode evoluir de um modelo estático para uma arquitetura dinâmica, capaz de se reconfigurar em tempo real de acordo com o comportamento dos usuários e a carga da aplicação.

A Cell-Based Architecture representa um avanço significativo na engenharia de sistemas distribuídos, oferecendo um modelo robusto, escalável e resiliente, especialmente adequado para ambientes de nuvem pública e híbrida. Sua fundamentação teórica, inspirada no padrão bulkhead, e suas aplicações práticas demonstram sua viabilidade técnica e seu potencial estratégico. No entanto, a complexidade associada à sua implementação exige maturidade arquitetural, domínio sobre práticas de automação e monitoramento, e um esforço coordenado para mitigar os riscos operacionais inerentes a arquiteturas fortemente distribuídas.

3 DISCUSSÃO DOS RESULTADOS

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

Como resultado dessa pesquisa, discute-se criticamente os achados da revisão bibliográfica em torno da aplicação da Cell-Based Architecture (CBA) na computação em nuvem, organizando os resultados em três frentes analíticas: impactos operacionais e de desempenho, desafios técnicos e limitações estruturais, e tendências emergentes associadas à evolução do modelo celular. Ao longo das subseções, serão exploradas evidências que demonstram ganhos expressivos em termos de resiliência, isolamento de falhas e escalabilidade seletiva, ao mesmo tempo em que se discutem os obstáculos relacionados à complexidade operacional, consistência de dados e gestão de infraestrutura. Também serão abordadas inovações recentes, como a reorganização adaptativa de células, integração com malhas de serviço e propostas de métricas para avaliação de eficácia arquitetural. Essas análises visam construir uma compreensão abrangente e crítica da viabilidade, maturidade e potencial futuro da CBA como modelo arquitetônico dominante em sistemas distribuídos modernos.

3.1 Impactos da Arquitetura Celular na Resiliência e Escalabilidade de Sistemas em Nuvem

Esta subseção discute os efeitos da Cell-Based Architecture (CBA) sobre a resiliência e a escalabilidade em sistemas distribuídos implementados em ambientes de computação em nuvem. Serão analisadas evidências provenientes de estudos de caso, documentação técnica e literatura científica que apontam para a mitigação de falhas sistêmicas por meio de isolamento estrutural, a capacidade de escalar seletivamente domínios com alta demanda e os efeitos operacionais dessas práticas sobre métricas como

REVISTA TÓPICOS

MTTR, MTBF e tempo de resposta sob carga. Também se discutirá como a granularidade das células influencia a elasticidade da infraestrutura e como essas propriedades contribuem para arquiteturas mais robustas e sustentáveis. Ao final, será apresentada uma avaliação crítica das condições necessárias para que os benefícios da CBA se concretizem de forma consistente e segura.

A Cell-Based Architecture emerge como uma resposta arquitetônica aos desafios impostos pela complexidade crescente dos sistemas distribuídos modernos. Sua principal contribuição reside na capacidade de isolar falhas em domínios limitados, o que reduz significativamente o impacto sistêmico de interrupções. Segundo Pisani e Gancarz (2024), a fragmentação do sistema em células autônomas permite que uma falha em um domínio específico, como um banco de dados indisponível ou uma sobrecarga localizada, não afete os demais usuários do sistema. Essa contenção do chamado blast radius se traduz em aumento da confiabilidade e da disponibilidade do serviço como um todo.

A resiliência é, portanto, diretamente impactada pelo desenho da arquitetura celular. No caso documentado pela AWS (2024), cada célula é projetada para ser completamente isolada, com seus próprios serviços, banco de dados, métricas e controle de acesso. A falha de uma célula não compromete as demais, garantindo continuidade da operação global. Essa abordagem foi adotada com sucesso por empresas como Slack e DoorDash. No caso do Slack, a segmentação da base de usuários em células por zonas de disponibilidade garantiu que incidentes de infraestrutura afetassem apenas

REVISTA TÓPICOS

um subconjunto específico da aplicação. De forma semelhante, o projeto SuperCell da DoorDash criou réplicas completas da aplicação para diferentes domínios, permitindo atualizações, testes e recuperação de falhas com impacto restrito (DZone, 2024; Rackspace, 2024).

Além de conter falhas, a CBA melhora a capacidade de recuperação. A métrica MTTR, amplamente usada na engenharia de confiabilidade, apresentou redução significativa nos estudos analisados. Conforme relatado por Gupta (2025), sistemas que adotaram CBA reduziram em até 35 por cento o tempo de restauração em comparação com arquiteturas baseadas em microsserviços tradicionais. Isso se deve à atuação localizada das equipes de operação e ao escopo reduzido dos incidentes. Em contrapartida, a métrica MTBF, que representa o tempo médio entre falhas, aumentou em função da menor complexidade e do isolamento funcional das células. Segundo dados apresentados por AWS Re:Invent (2024), o MTBF em ambientes CBA superou os de arquiteturas monolíticas em até 28 por cento, quando associada a práticas de automação e monitoramento proativo.

A escalabilidade também é profundamente impactada pela adoção da CBA. Em arquiteturas convencionais, o escalonamento horizontal é realizado por replicação de serviços inteiros, o que nem sempre é eficiente. A CBA permite escalonamento seletivo, célula por célula, de acordo com o perfil de uso e carga de trabalho. Isso significa que células com maior demanda podem receber mais recursos computacionais, enquanto células estáveis mantêm sua configuração mínima. Conforme análise do InfoQ (2024), essa granularidade de escalabilidade pode reduzir em até 25 por cento o consumo

REVISTA TÓPICOS

de recursos em comparação com modelos homogêneos de escalonamento horizontal, além de proporcionar economia significativa em custos operacionais.

Estudos de caso reforçam essa perspectiva. A AWS (2023) documentou que ambientes estruturados com CBA atingiram níveis mais altos de elasticidade com menor desperdício de infraestrutura, graças à segmentação da carga de trabalho. Já no relatório técnico da Rackspace (2024), observa-se que a escalabilidade segmentada permitiu otimização na alocação de memória, CPU e armazenamento, alinhando as células às suas reais necessidades de desempenho. Esse modelo também favorece a resiliência operacional, pois evita que picos de demanda em uma célula afetem negativamente outras partes do sistema.

Do ponto de vista organizacional, a arquitetura celular permite uma governança distribuída mais eficiente. Cada célula pode ser mantida por uma equipe autônoma, responsável por seu ciclo de vida, atualização, segurança e monitoramento. Isso facilita a adoção de práticas de DevOps, com entregas contínuas mais seguras e rápidas. A AWS (2024) destaca que essa separação de responsabilidades reduz o tempo de resposta a incidentes e aumenta a qualidade das entregas, pois cada equipe conhece profundamente seu domínio específico. A descentralização da governança também reduz a sobrecarga sobre times centralizados e estimula a especialização técnica.

Porém, para que os benefícios de resiliência e escalabilidade se concretizem, a modelagem arquitetural precisa ser criteriosa. A definição do escopo e da granularidade de cada célula deve considerar aspectos como independência

REVISTA TÓPICOS

de estado, perfil de uso, distribuição geográfica e criticidade funcional. Segundo Costa e Mendonça (2025), algoritmos de detecção de comunidades aplicados a grafos de chamadas de serviço podem auxiliar na identificação de agrupamentos naturais, contribuindo para uma segmentação mais eficiente e coerente com os padrões reais de interação entre componentes. O estudo desses autores demonstrou que a realocação das células com base nesses agrupamentos reduziu o tempo médio de resposta em 18 por cento, ao mesmo tempo em que aumentou a resiliência frente a falhas em cascata.

Outro ponto crítico está na implantação do componente responsável pelo roteamento das requisições, o chamado cell router. Esse elemento central, embora não execute lógica de negócios, precisa operar com latência mínima, alta disponibilidade e redundância, já que seu funcionamento afeta diretamente o acesso a todas as células. A AWS recomenda a utilização de roteadores distribuídos, com cache local e mecanismos de fallback, para evitar que uma falha no roteador comprometa o desempenho geral da aplicação (AWS Re:Invent, 2024). A ausência de planejamento nesse aspecto pode introduzir pontos únicos de falha, justamente o que a arquitetura celular busca eliminar.

A consistência de dados entre células, ainda que não comprometa diretamente a escalabilidade ou a resiliência, pode afetar a experiência do usuário e a integridade das transações. Como as células são isoladas, os dados devem ser replicados ou sincronizados de forma assíncrona. Estratégias como event sourcing e CQRS são frequentemente utilizadas, mas requerem um modelo mental adequado e testes rigorosos para garantir que as

REVISTA TÓPICOS

inconsistências temporárias não resultem em falhas de lógica ou em duplicidade de ações. Conforme evidenciado no relatório da InfoQ (2024), aplicações que não consideram explicitamente o modelo de consistência eventual enfrentam dificuldades em garantir confiabilidade percebida e previsibilidade nas respostas.

A arquitetura celular, portanto, representa um avanço significativo no design de sistemas escaláveis e resilientes, mas sua implementação bem-sucedida depende de escolhas arquitetônicas e operacionais alinhadas. Os ganhos são evidentes quando o modelo é bem aplicado, conforme demonstram os estudos revisados. A possibilidade de isolar falhas, escalar recursos conforme a demanda real e organizar o desenvolvimento de forma distribuída fornece às organizações uma vantagem substancial em cenários de alta complexidade. No entanto, essa vantagem só é concretizada quando há maturidade tecnológica, automação robusta e modelagem arquitetônica precisa.

3.2 Desafios Técnicos e Operacionais

Esta subseção examina criticamente os principais desafios técnicos e operacionais enfrentados na implementação da Cell-Based Architecture (CBA) em ambientes de computação em nuvem. São analisadas as limitações práticas relacionadas à replicação de infraestrutura, à gestão do roteamento intercelular, à consistência de dados e à complexidade organizacional exigida. Os achados foram extraídos de fontes técnicas como AWS, InfoQ, DZone e de estudos recentes que relatam experiências de

REVISTA TÓPICOS

adoção em larga escala, como nos casos do Slack, DoorDash e implementações em clusters Kubernetes.

A adoção da arquitetura celular pressupõe a replicação de diversos componentes de infraestrutura para cada célula. Isso inclui não apenas os microserviços, mas também bancos de dados, caches, sistemas de observabilidade, ferramentas de autenticação, pipelines de integração contínua e serviços de mensageria. Essa duplicação é necessária para manter o isolamento entre células, mas acarreta um custo operacional elevado e exige automação robusta para provisionamento, orquestração e monitoramento. Segundo o AWS Well-Architected Framework (2024), a criação de uma célula envolve o provisionamento de uma stack completa, o que só se torna viável com pipelines de automação bem definidos e integrados ao controle de versão da infraestrutura. Na ausência dessas práticas, a manutenção manual das células torna-se impraticável, especialmente em ambientes com dezenas ou centenas de instâncias.

A complexidade operacional é ainda mais acentuada pela necessidade de observabilidade individualizada por célula. Cada unidade deve possuir dashboards, alertas, logs e métricas específicas, o que amplia o volume de dados coletados e exige maior esforço para análise e correlação de eventos. Conforme destacado por Pisani e Gancarz (2024), uma falha em uma célula deve ser identificada e tratada de forma isolada, sem afetar as demais, o que demanda um sistema de monitoramento distribuído e granular. A falta de padronização nas métricas pode levar a ruídos na comunicação entre equipes, dificultando a resposta a incidentes.

REVISTA TÓPICOS

Outro desafio estrutural refere-se ao cell router, responsável por encaminhar as requisições dos usuários à célula apropriada com base em uma chave de partição. Embora sua função seja limitada ao roteamento, sua criticidade é elevada, pois representa o ponto inicial de contato com o sistema. Uma falha nesse componente pode comprometer o acesso a múltiplas células, o que viola o princípio do isolamento. Segundo AWS (2023), é necessário implantar roteadores redundantes por região e configurar políticas de fallback para manter a disponibilidade em caso de falhas. Além disso, a lógica de roteamento precisa ser eficiente, de baixa latência e capaz de operar sob alta concorrência, o que exige atenção à arquitetura de redes e ao escalonamento horizontal do próprio roteador.

A consistência de dados entre células constitui outro desafio central. Em muitos cenários, a lógica de negócio requer a troca de informações entre células ou a sincronização de estados. No entanto, a manutenção de uma consistência forte entre unidades isoladas compromete os benefícios do isolamento. Por isso, a maioria das aplicações baseadas em CBA adota modelos de consistência eventual, com replicação assíncrona de eventos ou atualização periódica de dados. Estratégias como event sourcing, change data capture e CQRS são frequentemente utilizadas, mas aumentam a complexidade da lógica de aplicação. Segundo InfoQ (2024), muitos times subestimam o impacto dessas decisões, resultando em inconsistências perceptíveis ao usuário final ou em duplicidade de operações em sistemas financeiros.

REVISTA TÓPICOS

A definição do tamanho e da granularidade das células também impõe decisões difíceis. Células muito pequenas ampliam o isolamento, mas aumentam o número de instâncias a serem gerenciadas, o que pode elevar a latência de roteamento e a sobrecarga de rede. Por outro lado, células muito grandes reduzem a complexidade operacional, mas comprometem a contenção de falhas. O estudo de Gupta (2025) em arquiteturas Kubernetes mostra que células com média de cinco microserviços apresentaram melhor equilíbrio entre performance e isolamento. No entanto, esses resultados variam conforme o domínio de negócio, a distribuição geográfica dos usuários e o perfil de carga. Portanto, não há uma configuração universal ideal, sendo necessário realizar testes e simulações para encontrar o ponto ótimo para cada sistema.

A própria cultura organizacional pode se tornar uma barreira à adoção da CBA. A arquitetura pressupõe a existência de times multidisciplinares com autonomia para operar cada célula. Isso exige maturidade em práticas DevOps, clareza na divisão de responsabilidades e forte alinhamento entre equipes. Conforme observado por Mohanagandhi e Ramalingam (2024), empresas que implementaram a CBA em ambientes Kubernetes enfrentaram dificuldades na padronização de processos entre células, especialmente quando diferentes equipes possuíam níveis distintos de conhecimento técnico ou seguiam práticas não homogêneas de versionamento e integração contínua. A governança distribuída, embora vantajosa, impõe riscos de fragmentação operacional se não for sustentada por diretrizes claras e mecanismos de auditoria e controle.

REVISTA TÓPICOS

Outro obstáculo identificado está na carência de ferramentas nativas que suportem diretamente o modelo de CBA. Embora orquestradores como Kubernetes e plataformas como AWS forneçam os elementos necessários para sua construção, não existem ferramentas de mercado amplamente consolidadas que ofereçam suporte direto ao ciclo de vida de células. A automação do cell router, a gestão de quotas, a criação de malhas de serviços e a replicação segura de dados ainda dependem de soluções customizadas, o que eleva a barreira de entrada e exige investimento significativo em engenharia interna.

Por fim, a avaliação do sucesso de uma arquitetura baseada em células carece de métricas padronizadas. Embora indicadores como MTTR e MTBF sejam úteis, eles não capturam aspectos específicos da operação segmentada. Costa e Mendonça (2025) sugerem a criação de novos indicadores, como o índice de isolamento efetivo, que mede a capacidade do sistema de conter falhas dentro de uma única célula, e o grau de acoplamento intercelular, que indica a frequência de chamadas entre células. Tais métricas poderiam orientar melhorias contínuas e fornecer dados empíricos para justificar o investimento na arquitetura celular.

Dessa forma, os desafios técnicos e operacionais da Cell-Based Architecture são proporcionais aos benefícios que ela oferece. A segmentação do sistema em células independentes melhora a resiliência e a escalabilidade, mas exige automação, monitoramento e governança de alto nível. A adoção bem-sucedida do modelo depende da capacidade das organizações em lidar com a complexidade estrutural e organizacional, integrar ferramentas eficazes e

REVISTA TÓPICOS

formar equipes capacitadas para operar com autonomia e consistência. Os estudos analisados demonstram que, embora a curva de adoção seja acentuada, os ganhos em robustez e controle justificam os esforços, desde que acompanhados por um planejamento arquitetural criterioso e uma cultura de engenharia avançada.

3.3 Tendências Emergentes e Perspectivas Futuras

Esta subseção analisa as tendências emergentes relacionadas à evolução da Cell-Based Architecture (CBA), com base em estudos técnicos e acadêmicos que apontam para caminhos de aprimoramento da automação, da inteligência arquitetural e da integração com novas tecnologias como service mesh, machine learning e redes adaptativas. Serão discutidas também as perspectivas para a aplicação da CBA em contextos de autoescalamento dinâmico, reorganização automática de células, observabilidade integrada e desenvolvimento de métricas específicas para avaliação da eficácia do isolamento. Ao final, são apresentadas recomendações para pesquisas futuras que busquem consolidar a maturidade do modelo arquitetural celular.

A maturidade da Cell-Based Architecture como modelo arquitetônico depende diretamente da capacidade das organizações de adaptá-la às demandas de sistemas dinâmicos, geograficamente distribuídos e orientados a eventos. Estudos recentes, como o de Costa e Mendonça (2025), têm explorado o uso de algoritmos de detecção de comunidades para otimizar a organização das células com base em padrões reais de comunicação entre serviços. Nessa abordagem, grafos de chamadas são analisados por algoritmos de modularidade que identificam agrupamentos naturais entre

REVISTA TÓPICOS

componentes, permitindo reorganizações arquitetônicas que favorecem a coesão interna das células e minimizam a necessidade de chamadas externas. Essa técnica foi testada em ambientes simulados e resultou em redução de 18 por cento no tempo médio de resposta, além de aumentar a resiliência frente a falhas de rede e gargalos operacionais.

Outra tendência promissora está na integração entre CBA e service mesh, uma camada de infraestrutura que gerencia a comunicação entre microserviços com recursos nativos de segurança, observabilidade e controle de tráfego. Ferramentas como Istio, Linkerd e Consul vêm sendo adaptadas para ambientes celulares, permitindo que políticas como retries, circuit breakers, rate limits e roteamento por contexto sejam aplicadas diretamente entre células, sem necessidade de configuração explícita nos serviços. Segundo o relatório técnico da InfoQ (2024), essa integração facilita o monitoramento e o balanceamento de carga entre células, além de simplificar a gestão da segurança e da criptografia de dados em trânsito. A observabilidade também é beneficiada, com métricas como latência, taxa de erro e disponibilidade sendo extraídas diretamente da malha de serviços, sem necessidade de instrumentação adicional.

A aplicação de inteligência artificial e aprendizado de máquina no gerenciamento da arquitetura celular também tem ganhado espaço. Modelos preditivos vêm sendo utilizados para antecipar picos de carga, identificar padrões anômalos de tráfego e sugerir realocações de serviços entre células. Em ambientes que operam com centenas de células e milhões de requisições por segundo, a capacidade de reagir de forma inteligente a variações

REVISTA TÓPICOS

operacionais torna-se um diferencial estratégico. A AWS (2024) tem explorado modelos de autoescalamento baseados em aprendizado, nos quais as células ajustam sua capacidade de forma autônoma com base em previsões de uso, reduzindo a latência de resposta e otimizando o uso de recursos computacionais.

A reorganização dinâmica de células é outra linha de evolução que tem despertado interesse. Em vez de manter uma estrutura fixa de particionamento, a arquitetura celular pode se reorganizar periodicamente com base na análise de logs, métricas e eventos. Essa reorganização visa redistribuir a carga de forma mais eficiente, minimizar interdependências e adaptar a topologia do sistema à realidade operacional. Segundo Pisani e Gancarz (2024), essa abordagem requer mecanismos de versionamento de células, ferramentas de migração de estado e roteadores com capacidade de reconfiguração dinâmica. A combinação desses elementos abre caminho para arquiteturas adaptativas que reagem ao comportamento dos usuários e às mudanças na infraestrutura de forma autônoma.

Além da evolução tecnológica, as perspectivas futuras da CBA dependem da criação de métricas específicas para avaliar sua eficácia. As métricas tradicionais de disponibilidade, como MTTR e MTBF, são importantes, mas não capturam a totalidade do comportamento de uma arquitetura celular. Costa e Mendonça (2025) propuseram dois indicadores adicionais: o índice de isolamento, que mede a capacidade de uma falha ser contida dentro de uma única célula, e o índice de coesão intercelular, que avalia a frequência e a intensidade das comunicações entre células. Esses indicadores permitem

REVISTA TÓPICOS

avaliar se o particionamento está bem calibrado e se a arquitetura está cumprindo seu papel de modularização eficiente e contenção de impacto.

O uso de ambientes multi-cloud e arquiteturas híbridas também se apresenta como uma fronteira para a expansão da CBA. Em cenários onde diferentes partes do sistema operam em provedores distintos, a arquitetura celular permite segmentar responsabilidades por ambiente, mantendo isolamento físico e lógico. Essa abordagem é particularmente útil em aplicações que exigem compliance com regulamentações específicas, como LGPD ou GDPR, permitindo que dados sensíveis sejam processados em regiões geograficamente controladas sem comprometer a integridade da aplicação como um todo. Segundo o AWS (2024), a separação de células por ambiente facilita a governança de políticas de segurança, auditoria e replicação de dados.

Por fim, há um movimento em direção à criação de frameworks e plataformas que suportem a adoção de CBA de forma nativa. Atualmente, a implementação da arquitetura celular depende de uma combinação de ferramentas, scripts e automações desenvolvidas internamente. No entanto, iniciativas como CellMesh, Cellula e Mosaic buscam oferecer plataformas de orquestração, roteamento, monitoramento e segurança especificamente desenhadas para arquiteturas celulares. Essas ferramentas visam reduzir a barreira de entrada e acelerar o processo de adoção, fornecendo templates, APIs e práticas recomendadas para o desenho, implantação e operação de células.

REVISTA TÓPICOS

As tendências emergentes e as perspectivas futuras da Cell-Based Architecture apontam para um modelo arquitetônico em constante evolução, que busca responder aos desafios da complexidade sistêmica com modularidade, automação e inteligência. A integração com tecnologias como service mesh, aprendizado de máquina e análise de grafos, aliada à criação de métricas específicas e frameworks dedicados, sugere que a CBA poderá se consolidar como um padrão dominante para sistemas distribuídos de larga escala. No entanto, sua consolidação dependerá da maturidade das ferramentas, da padronização dos processos e do desenvolvimento de conhecimento técnico aprofundado por parte das equipes de engenharia de software. O investimento em pesquisa aplicada, testes em produção e compartilhamento de experiências entre organizações será decisivo para transformar a CBA de um modelo promissor em uma prática consolidada na indústria de software.

4 CONCLUSÃO

A Cell-Based Architecture (CBA) representa uma resposta arquitetônica robusta aos desafios contemporâneos enfrentados por sistemas distribuídos em ambientes de computação em nuvem. Ao longo deste estudo, constatou-se que sua adoção proporciona melhorias significativas em resiliência, escalabilidade e governança descentralizada, atributos indispensáveis para aplicações que operam sob alta demanda, variabilidade de carga e exigência contínua de disponibilidade. Fundamentada na separação lógica e funcional de sistemas em células autônomas, a CBA possibilita a contenção eficiente

REVISTA TÓPICOS

de falhas, a segmentação do tráfego e a realização de deploys seletivos, elevando os níveis de confiabilidade operacional e performance.

A análise sistemática da literatura técnica e científica demonstrou que a arquitetura celular permite reduzir o tempo médio de recuperação de falhas (MTTR), aumentar o tempo médio entre falhas (MTBF) e realizar escalonamentos seletivos por domínio, o que contribui para a utilização mais eficiente dos recursos computacionais. Casos práticos documentados em grandes empresas, como Slack e DoorDash, exemplificam como a segmentação da aplicação em células pode mitigar falhas em cascata, otimizar operações de DevOps e reduzir custos operacionais. Essas constatações são corroboradas por métricas empíricas extraídas de estudos conduzidos por AWS, InfoQ, Rackspace e autores independentes.

Contudo, a adoção da CBA impõe desafios técnicos e organizacionais substanciais. A replicação de infraestrutura, a complexidade do roteamento entre células, a necessidade de consistência eventual de dados e a maturidade dos times de engenharia são fatores que elevam a barreira de entrada para sua implementação. A arquitetura celular exige automação avançada, observabilidade granular, ferramentas especializadas e um modelo organizacional compatível com o paradigma de governança distribuída. Esses aspectos tornam imprescindível a realização de um planejamento arquitetônico rigoroso e a estruturação de equipes com domínio sobre práticas modernas de desenvolvimento, integração contínua e SRE (Site Reliability Engineering).

REVISTA TÓPICOS

As tendências emergentes apontam para uma evolução da CBA em direção à inteligência arquitetural. Iniciativas como o uso de algoritmos de detecção de comunidades, integração com service mesh, aplicação de machine learning para autoescalonamento e reorganização adaptativa das células revelam um movimento em direção a arquiteturas autônomas e responsivas. O desenvolvimento de métricas específicas para avaliação do isolamento e da coesão entre células reforça a necessidade de uma abordagem quantitativa e orientada a dados na gestão dessas arquiteturas. Além disso, o surgimento de frameworks dedicados ao suporte da CBA sugere que, em breve, essa abordagem poderá se consolidar como padrão para sistemas distribuídos de alta complexidade.

Dessa forma, conclui-se que a Cell-Based Architecture apresenta elevado potencial para se tornar uma arquitetura de referência na computação em nuvem, desde que acompanhada por práticas maduras de engenharia e apoiada por ferramentas e processos adequados. Seus benefícios são evidentes, mas sua aplicação exige discernimento técnico, análise de domínio e constante adaptação. A pesquisa futura deve focar na sistematização de boas práticas, no desenvolvimento de ferramentas de suporte nativo, na definição de padrões abertos e na avaliação empírica em diferentes contextos organizacionais e setoriais. Ao articular avanços técnicos com estratégias de implementação sustentáveis, a comunidade científica e a indústria poderão explorar plenamente o potencial da CBA como um marco evolutivo na arquitetura de sistemas distribuídos.

REFERÊNCIAS BIBLIOGRÁFICAS

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

AWS. (2023). Well-Architected Framework – Cell-Based Architecture best practices. AWS Documentation.

AWS. (2024). Cell-Based Architecture on AWS: Reducing Blast Radius. AWS Solutions Library.

AWS re:Invent. (2024). ARC312: Using cell-based architectures for resilient design on AWS. Apresentação em conferência.

Cortex. (2024). Your guide to incident response metrics: MTTR, MTBF, MTTF explained. Cortex Blog. Recuperado de <https://www.cortex.io>

Costa, F. R., & Mendonça, L. T. (2025). Aplicação de algoritmos de detecção de comunidades para otimização de arquiteturas celulares. Anais da Conferência Brasileira de Engenharia de Software.

DZone. (2024). Grokking cell-based architecture: Scalable and resilient cloud design. DZone Guide.

Fowler, M., & Lewis, J. (2014). Microservices: a definition of this new architectural term. MartinFowler.com.

Gupta, A. (2025). Cell-based architecture for scalable cloud systems: A Kubernetes case study. Journal of Cloud Computing Practice, 12(1), 45–62.

Mell, P., & Grance, T. (2011). The NIST definition of cloud computing (Special Publication 800-145). National Institute of Standards and Technology.

REVISTA TÓPICOS - ISSN: 2965-6672

REVISTA TÓPICOS

Mohanagandhi, V., & Ramalingam, G. (2024). Implementing cell-based architecture in Kubernetes for e-commerce systems. *International Journal of Software Research and Practice*, 18(4), 123–139.

Pisani, E., & Gancarz, R. (2024). How cell-based architecture enhances modern distributed systems. *InfoQ*. Recuperado de <https://www.infoq.com>

Rackspace Technology. (2024). Cell-based architecture strategies and patterns. *Rackspace Blog*. Recuperado de <https://www.rackspace.com>

Silva, J. R., & Almeida, F. T. (2025). Optimization of cell-based architecture using community detection techniques. *Proceedings of the International Conference on Distributed Systems Architecture*, 22(3), 101–118.

System Design Newsletter. (2024). Balancing cell granularity in cell-based architecture. *System Design One*.

Tharmarajah, N. (2024). Cell-Based Architecture on AWS: Practical Approaches to Fault Isolation and Scalability.

¹ Bacharel em Engenharia de Produção pela Faculdade Estácio do Recife, Doctor in Business Administration pela Atlanta College of Liberal Arts and Sciences e Estudante em Master of Science in Business Administration pela Must University. E-mail: engcfraposo@outlook.com.br