Composable ERP: a próxima geração de arquitetura de sistemas corporativos

Em grandes empresas, existe uma verdade estrutural sobre paisagem de aplicações que tem se tornado incômoda de admitir. Análises setoriais indicam que a empresa média moderna opera aproximadamente 975 aplicações SaaS em produção, e a maioria dessas aplicações não consegue compartilhar dados de forma nativa, não pode ser substituída sem reescrita substancial e está travada em ciclos de upgrade definidos pelo fornecedor que têm pouco a ver com o roadmap real do negócio. Ao lado dessas centenas de SaaS, coexiste tipicamente uma stack tradicional de ERP monolítico (SAP, Oracle, outros) que sustenta funções core mas cuja rigidez arquitetural limita drasticamente a velocidade com que a empresa pode responder a mudanças de mercado, incorporar tecnologias emergentes ou entrar em novos negócios. Cada tentativa de evoluir uma capacidade específica esbarra em impacto colateral em dez outras dependências, cada upgrade da suíte principal demanda meses de teste de regressão, cada nova integração exige middleware customizado que amanhã será dívida técnica. É exatamente esse impasse estrutural que a arquitetura de **Composable ERP** foi construída para resolver, oferecendo modelo em que sistemas corporativos são montados a partir de capacidades de negócio modulares, substituíveis e conectadas por APIs, permitindo que cada componente evolua de forma autônoma sem forçar coordenação com todos os outros. Para gerentes e diretores de TI em empresas que reconheceram que continuar apostando em ERPs monolíticos como resposta única não sustenta a velocidade que o negócio exige, entender essa arquitetura virou tema de decisão executiva com implicações diretas em agilidade, custo de mudança e capacidade de participar da onda de automação agêntica.

Vale começar com contexto honesto. O termo **Composable ERP** foi cunhado por analistas do Gartner em 2020, inicialmente como parte da tese mais ampla de "Composable Enterprise" desenvolvida por Yefim Natis e Gene Alvarez. A ideia central reenquadrou o ERP tradicional (assim como CRM e MES) como artefato das condições econômicas de software empacotado dos anos 1990, e não como característica intrínseca do trabalho corporativo que precisa ser automatizado. Análises do Gartner apontam que aproximadamente 70% das organizações adotarão tecnologia composable até o final de 2026, com early adopters relatando 80% mais velocidade em deployment de novas features e até 5% de crescimento adicional de receita comparado a organizações presas em stacks monolíticas. Esse cenário conversa diretamente com o que abordamos no conteúdo sobre otimização estratégica de TI, porque a decisão de continuar operando sobre arquitetura monolítica ou migrar progressivamente para modelo composable é hoje uma das discussões arquiteturais mais impactantes que uma diretoria de TI enfrenta.

O que é Composable ERP, e o problema estrutural que ele resolve

A definição precisa importa. **Composable ERP** é o padrão arquitetural de construir sistemas corporativos a partir de componentes modulares, substituíveis e conectados por APIs, em vez de implementar uma suíte monolítica única de um fornecedor. Cada capacidade de negócio (order management, pricing, inventory, financial accounting, procurement, subscription billing, entre outras) é encapsulada como serviço autônomo com interface bem definida, e a composição desses serviços em processos de negócio acontece via camada de orquestração desenhada explicitamente para essa função. Empresas que operam com **Composable ERP** podem substituir uma capacidade específica sem reescrever todo o sistema, incorporar best-of-breed de múltiplos fornecedores em domínios diferentes, e evoluir cada componente em cadência própria.

Cinco elementos distinguem a proposta de valor de forma concreta. Primeiro, modularity real onde cada capacidade é independentemente deployable e encapsulada, sem shared state escondido entre componentes. Segundo, autonomy operacional onde cada capacidade gerencia seu próprio ciclo de vida sem precisar coordenar releases com outras. Terceiro, orchestration explícita via camada de coordenação que conecta capacidades em processos de negócio sem embutir lógica de negócio na camada de integração. Quarto, vendor-agnosticism que permite montar a stack combinando fornecedores diferentes conforme cada domínio, evitando lock-in em um único vendor para tudo. Quinto, adaptabilidade contínua que permite ao negócio experimentar, ajustar e recompor capacidades conforme condições mudam, sem projetos massivos de replatforming.

Vale separar com clareza o que **Composable ERP** é do que ele não é. Ele não é sinônimo de "ERP modular", termo que tipicamente descreve sistemas monolíticos organizados em módulos logicamente separados mas com database compartilhado e ciclo de deployment único. Ele não é apenas microserviços; migração para microserviços que produz dezenas de serviços técnicos com mesma coordenação que o monolito original é falha comum. Ele não é bandeira anti-SAP ou anti-Oracle; muitas implementações mantêm ERP tradicional para finanças core e adotam composabilidade em domínios customer-facing ou operacionais. Ele não é aquisição de bala de prata; adoção madura exige governança sofisticada de APIs, disciplina arquitetural forte e maturidade organizacional para gerenciar múltiplos fornecedores em vez de um único throat to choke. Esse ponto conversa diretamente com o que discutimos em Enterprise Architecture, porque disciplina arquitetural madura é pré-requisito operacional para que **Composable ERP** entregue benefício em vez de virar caos gerenciado.

Os princípios MACH que sustentam Composable ERP

O framework técnico dominante para materializar **Composable ERP** é conhecido como MACH, acrônimo cunhado por commercetools em 2020 e institucionalizado pela MACH Alliance, organização sem fins lucrativos que agrega vendors, integradores e end-users comprometidos com esse padrão. MACH descreve quatro propriedades arquiteturais que, juntas, definem sistemas

verdadeiramente composable, e sistema que falha em qualquer uma delas não é composable no sentido forte, independente do que o marketing diga.

O primeiro princípio é **Microservices**: aplicações construídas como coleções de serviços pequenos, independentemente deployables, alinhados a capacidades específicas de negócio em vez de camadas técnicas. O segundo é **API-first**: toda funcionalidade exposta através de APIs estáveis e bem documentadas, com UI sendo apenas um consumidor entre muitos. O terceiro é **Cloud-native**: construído para infraestrutura cloud, aproveitando escalabilidade elástica e serviços gerenciados, não simplesmente hospedado em cloud. O quarto é **Headless**: separação clara entre front-end e back-end, permitindo flexibilidade máxima na experiência do usuário e habilitando múltiplos front-ends (web, mobile, kiosk, agents) contra o mesmo backend.

Vale registrar honestamente que o mercado sofre com fenômeno chamado "composable-washing", onde vendors adotam vocabulário composable e MACH sem entregar verdadeiramente as propriedades técnicas correspondentes. Certificação independente pela MACH Alliance existe justamente para separar plataformas que cumprem o padrão de plataformas que apenas se autodeclaram composable. Para referência canônica sobre os princípios e a evolução do padrão, vale conhecer diretamente a documentação oficial da MACH Alliance, fonte primária da terminologia amplamente adotada pela indústria e mantida com atualizações regulares.

Os três pilares Gartner: Modularity, Autonomy e Orchestration

Complementarmente ao framework MACH, o Gartner formalizou em 2020 três princípios arquiteturais que definem sistemas composable maduros. O primeiro é **Modularity**, propriedade em que capacidades são independentemente deployables e encapsuladas, com estado próprio, ciclo de vida próprio e interface bem definida. Modularity real significa que trocar uma capacidade por outra do mesmo tipo não deve exigir mudanças cascata em outras capacidades. Sistemas com "shared database" implícito ou dependências opacas entre módulos falham nesse critério mesmo quando apresentados como modulares.

O segundo pilar é **Autonomy**, propriedade em que cada capacidade gerencia seu próprio ciclo de vida sem precisar coordenar releases, upgrades ou deployment com outras capacidades. Autonomy real é o que permite que times de produto diferentes trabalhem em paralelo sem gargalos de coordenação, e é o que sustenta ganhos documentados de 40% a 60% em time-to-add de novas features. Sistemas que exigem "release trains" mensais coordenados entre múltiplos serviços falham nesse critério.

O terceiro pilar é **Orchestration**, camada de coordenação que conecta capacidades em processos de negócio sem embutir lógica de negócio na camada de integração. Orchestration madura mantém a lógica de negócio dentro dos serviços apropriados e usa a camada de integração apenas para roteamento, transformação de formato e correlação de eventos. Sistemas que acabam concentrando

regras de negócio críticas no middleware repetem o problema do monolito, apenas movido para camada diferente. Esse desenho conversa diretamente com o que discutimos em SAP BTP, porque plataformas modernas de integração e extensão são justamente o veículo natural para materializar orquestração composable dentro e ao redor do ecossistema SAP.

Packaged Business Capabilities: os blocos de construção do Composable ERP

Uma contribuição conceitual importante do Gartner para viabilizar **Composable ERP** foi a definição de **Packaged Business Capabilities (PBCs)**, blocos de construção que operacionalizam o conceito na prática. Um PBC é uma capacidade de negócio discreta exposta como serviço com APIs estáveis, com ownership claro, interfaces documentadas e ciclos de deployment independentes. Exemplos concretos incluem PBC para catálogo de produto, PBC para roteamento de ordem, PBC para detecção de fraude, PBC para billing de subscription, PBC para gestão de crédito, PBC para consolidação financeira, PBC para cálculo de impostos.

A diferença entre PBC e microserviço técnico é fundamental para entender por que composable funciona quando funciona. PBCs são business-aligned, não technical-aligned. Um PBC de "gestão de crédito" corresponde a uma capacidade de negócio que um executivo reconhece e pode discutir, enquanto um microserviço de "auth-service" ou "notification-queue" é decomposição técnica que não conversa com linguagem de negócio. Empresas que confundem os dois níveis acabam com dezenas de microserviços técnicos que exigem mesma coordenação do monolito original, sem os benefícios da composabilidade real. Esse ponto conversa com o que abordamos em gestão de processos empresariais, porque desenho de PBC bem feito começa com mapeamento sólido de capacidades e processos, não com decomposição arbitrária de código legado.

Adicionalmente, cada PBC precisa ter três atributos operacionais para funcionar em produção. Precisa ter ownership formal, com time responsável pela evolução, sustentação e roadmap. Precisa ter contrato de API estável, versionado explicitamente, com deprecation policy clara. Precisa ter observabilidade adequada, com métricas de negócio, SLOs, tracing e logging estruturado. PBC sem esses atributos vira liability arquitetural mesmo quando tecnicamente correto.

A jornada de adoção: Strangler Pattern e evolução progressiva

A questão prática que quase toda diretoria de TI enfrenta ao considerar **Composable ERP** é como sair do ponto atual, tipicamente dominado por ERP monolítico legado, para o modelo composable sem risco existencial. A resposta dominante em 2026 é o strangler pattern, popularizado por Martin Fowler e adotado em transformações de larga escala. A ideia central é preservar o ERP monolítico para funções core estáveis (tipicamente finanças central e contabilidade) enquanto capacidades específicas são progressivamente movidas para stack composable, começando por áreas onde velocidade de negócio é mais crítica.

Casos setoriais ilustram o padrão. A Coca-Cola European Partners (CCEP), um dos maiores bottling partners globais da Coca-Cola, adotou estratégia de **Composable ERP** para substituir seu sistema SAP monolítico anterior, usando abordagem modular com microserviços que permitiu integrar aplicações best-of-breed para finance, supply chain e operações de venda. Manufacturers têm adotado abordagens similares em MES (Manufacturing Execution Systems), com Composable MES emergindo como categoria própria. Setores B2B e B2C aplicam composabilidade em customer data platforms, POS, order management systems, DXP e adjacentes.

Três abordagens de sequenciamento merecem atenção. A primeira é **customer-facing first**, movendo capacidades ligadas a experiência do cliente (commerce, personalização, self-service) para composable enquanto backend transacional permanece monolítico. A segunda é **greenfield para novos negócios**, onde nova unidade de negócio ou geografia é construída composable desde o início, sem carga de legado. A terceira é **domain-by-domain migration**, com cada domínio de negócio migrado como programa próprio ao longo de anos. Esse tema dialoga diretamente com o que abordamos em modernização de aplicações legadas, porque **Composable ERP** é frequentemente o modelo alvo de programas maiores de modernização de aplicações em grandes empresas.

A evolução do Composable ERP em 2026: IA agêntica como driver

A frente que mais tem transformado a discussão sobre **Composable ERP** em 2026 é a chegada em escala de IA agêntica. Agentes autônomos que operam em múltiplos sistemas corporativos precisam de arquitetura composable para funcionar em escala. Sem APIs bem definidas, dados acessíveis semanticamente e capacidades encapsuladas como serviços, agentes ficam presos em pilotos isolados que entregam resultado localizado mas não escalam através do negócio. A MACH Alliance formalizou essa constatação em 2025, transitando para modelo de liderança end-user até setembro de 2026 e lançando iniciativas como MACH AI Exchange e Open Data Model justamente para acelerar transformação AI-ready.

Três padrões práticos merecem atenção nessa convergência. O primeiro é **AI-driven process optimization** materializada em serverless functions expostas como PBCs, permitindo que decisões automatizadas sejam plugadas em processos existentes sem reescrita de fluxo. O segundo é **agentic orchestration** onde agentes coordenam trabalho entre múltiplas capacidades sem que cada capacidade precise conhecer o agente, mantendo desacoplamento. O terceiro é **AI-powered PBCs** onde capacidades específicas incorporam modelos de IA como parte natural do serviço (fraud detection, credit scoring, demand forecasting, procurement recommendation). Esse tema dialoga diretamente com o que discutimos em governança de IA nas empresas, porque governança de agentes em escala depende de arquitetura corporativa que permita observabilidade, controle e substituibilidade de componentes de IA sem quebrar o resto da operação.

Onde o Composable ERP entrega valor concreto em grandes operações

Para uma diretoria que precisa priorizar investimento, vale mapear honestamente onde **Composable ERP** tem entregado retorno mais consistente. Cinco famílias de valor concentram o melhor histórico. A primeira é **aceleração drástica de time-to-market para novas features**. Empresas que atingiram nível 3 ou 4 de maturidade composable em pelo menos um domínio reportam reduções de 40% a 60% em tempo entre ideia e feature em produção, principalmente porque times de produto podem projetar, testar e implantar mudanças em seu PBC sem coordenar com múltiplos outros times.

A segunda família é **flexibilidade estratégica em decisões de vendor**. Ficar preso em vendor único com custos crescentes de migração é uma das armadilhas históricas mais custosas em TI corporativa; composable dá à empresa poder real de negociação e capacidade concreta de substituir componente específico sem trocar toda a stack. A terceira família é **habilitação de experimentação e inovação**. Empresas com stack composable podem testar novas capacidades em subset de clientes ou geografia sem risco cascata em outras áreas, criando cultura de experimentação estruturada que monolitos inibem.

A quarta família é **fundação para adoção de IA em escala**. Como discutido, agentes precisam de composabilidade para operar através do negócio. Empresas com fundação composable herdam vantagem estrutural em adoção de IA agêntica. A quinta família é **participação em ecossistemas e parcerias**. APIs bem desenhadas e capacidades expostas como serviços permitem que a empresa participe de ecossistemas digitais, integre com parceiros e crie novos modelos de negócio que exigem interoperabilidade que monolitos não entregam. Esse tema conversa com o que abordamos em SAP LeanIX, porque governança de portfólio de PBCs e de APIs em ambiente composable exige plataforma de EA management madura para não virar caos gerenciado.

Os erros mais comuns em adoção de Composable ERP

Falar de **Composable ERP** sem nomear honestamente os erros comuns seria desserviço. Cinco armadilhas concentram a maior parte dos casos onde o programa decepciona. A primeira é composable-washing, adotar vocabulário composable mas continuar operando essencialmente monolítico. Vendors dizem que suas plataformas são composable, arquitetos dizem que a stack é composable, mas na prática deployments continuam acoplados, APIs escondem shared state e coordenação entre times permanece necessária. Empresas maduras aplicam critérios rigorosos MACH para separar composable real de marketing.

A segunda armadilha é confundir microserviços técnicos com PBCs de negócio. Decompor código legado em vinte serviços técnicos que reproduzem o acoplamento original apenas move o problema para camada diferente. Empresas maduras começam com mapeamento de capacidades de negócio e derivam decomposição técnica a partir disso, não o contrário. A terceira armadilha é subestimar complexidade de governança em ambiente multi-vendor. Composable troca dependência de um vendor por dependência de muitos vendors, cada um com contratos, SLAs, roadmaps e

vulnerabilidades próprios. Empresas maduras estabelecem governança de vendor management robusta antes de expandir composabilidade.

A quarta armadilha é ignorar disciplina de API e data. **Composable ERP** vive ou morre com base na qualidade de APIs e na consistência semântica de dado entre serviços. Sem API-first cultura, sem versionamento explícito, sem governança de esquema, sem data mesh maduro, o resultado é caos integrado que combina o pior dos mundos. Esse ponto conversa diretamente com o que discutimos em SAP MDG, porque governança de dado mestre é fundação operacional para que serviços composable conversem semanticamente sem quebras. A quinta armadilha é big bang em busca de composabilidade total. Tentar transformar toda a paisagem em composable simultaneamente produz risco existencial e cronograma inatingível. Empresas maduras aplicam strangler pattern com paciência plurianual.

Indicadores que mostram se o programa está entregando valor

Programas sérios medem progresso em quatro dimensões. A primeira é **maturidade de composabilidade por domínio**. Aplicações do Composability Maturity Model medindo cada domínio contra cinco níveis (foundational, developing, defined, integrated, optimized), com progresso rastreado ao longo do tempo. A segunda dimensão é **velocidade de entrega**. Redução em time-to-market de novas features, aumento em frequência de deploy, redução em cross-team coordination overhead, aumento em quantidade de experimentos executados por período.

A terceira dimensão é **flexibilidade demonstrada**. Quantidade de PBCs substituídas ou reconfiguradas por período, quantidade de novos vendors integrados sem reescrita, quantidade de novos canais ou touchpoints habilitados. A quarta dimensão é **qualidade arquitetural**. Cobertura de APIs versionadas, aderência a padrões, número de dependências implícitas descobertas em incidentes, maturidade de observabilidade por PBC. Esse tema dialoga com o que abordamos em gestão de processos empresariais, porque disciplina de mensuração de processo é justamente o que sustenta melhoria contínua em ambiente composable maduro.

Como uma diretoria de TI deveria estruturar a adoção

A pergunta útil não é "vamos ser composable?", mas "como organizar a jornada para que ela transforme a arquitetura de sistemas de forma sustentável ao longo dos próximos três a cinco anos?". Quatro disciplinas costumam ser determinantes. A primeira é fazer assessment honesto do estado atual: mapear paisagem de aplicações, identificar domínios onde velocidade é mais crítica, avaliar maturidade organizacional para operar composable, entender apetite executivo por mudança arquitetural profunda. Sem esse retrato base, roadmap composable vira lista de desejos sem sequenciamento defensável.

A segunda disciplina é escolher domínio inicial com critérios claros. Domínio com valor de negócio alto para o cliente ou operação, com escopo delimitável, com sponsorship executivo claro e com apetite para novo modelo operacional costuma funcionar bem como piloto. Áreas customer-facing frequentemente pontuam bem em todos esses critérios. A terceira disciplina é estabelecer capacidades transversais que sustentam composabilidade em escala: plataforma de API management, service catalog, developer portal, observability platform, identity provider, event backbone, data mesh. Sem esses habilitadores, cada PBC precisaria construir capacidades duplicadas.

A quarta disciplina é integrar a estratégia composable com evolução mais ampla de SAP, cloud, IA e governança arquitetural. Adotar composabilidade sem alinhar com estratégia de SAP RISE ou SAP GROW para core ERP, com estratégia de dados, com padrões de segurança e com governança arquitetural produz resultado inferior à soma das partes. Esse tema dialoga com o que abordamos em AMS SAP, porque sustentação de ambientes híbridos que combinam SAP monolítico core com PBCs composable ao redor exige expertise específica que combina domínio SAP tradicional com fluência em arquiteturas cloud-native modernas.

Em última análise, **Composable ERP** é uma das transformações arquiteturais mais importantes que uma diretoria de TI pode conduzir quando a empresa reconhece que continuar apostando em ERPs monolíticos como resposta única já não sustenta a velocidade que o negócio exige. Quando adotada em modelo bem desenhado, com governance madura, roadmap por fase priorizando valor, disciplina de API e dados, e integração com o ecossistema mais amplo, ela transforma TI de custo reativo em capacidade estratégica, entrega agilidade que monolitos não conseguem, prepara a empresa para participar da onda de IA agêntica e cria fundação técnica para experimentar, evoluir e participar de ecossistemas digitais sem replatforming perpétuo. Quando tratada como buzzword de marketing, adotada superficialmente ou perseguida em busca de composabilidade total simultânea, produz risco existencial ou caos gerenciado que decepciona todo mundo.

Se a sua empresa quer estruturar a adoção de **Composable ERP** para modernizar a arquitetura de sistemas corporativos, ganhar flexibilidade estratégica, acelerar time-to-market e construir fundação preparada para IA agêntica e ecossistemas digitais, a Simple pode apoiar esse movimento com Arquitetura de Soluções, Mapeamento de Requisitos, Arquitetura de Software, Desenho de Soluções Completas, projetos com CELONIS, Consultoria e Execução SAP, Análise de Aderência, Implementação do S/4 Hana, Soluções Customizadas SAP, Integrações SAP com outros fornecedores e Terceirização de Tecnologia, incluindo busca, avaliação, alocação de profissionais e formação de squad. Entre em contato com a Simple para avaliar o estado atual da sua paisagem de aplicações e desenhar o caminho de evolução para Composable ERP que melhor se ajusta à realidade do seu negócio.