

E todos os lados seguem buscando disruptar o Cobol!

Primeiro foi o frenesi da migração para a cloud e agora o uso de GenAI para acelerar a conversão para Java.

Valendo destacar que nesse caso estamos falando de uma iniciativa da própria IBM com o seu Watson, segundo essa matéria da

InfoWorld:

https://www.infoworld.com/article/3705251/ibm-watsonx-to-use-generative-ai-to-translate-cobol-code-into-java.html

Na era digital em que vivemos, a modernização de sistemas legados é uma

necessidade imperativa para empresas que desejam manter-se competitivas e ágeis.

Neste contexto, a IBM, um gigante no campo da tecnologia, introduziu uma solução inovadora que promete revolucionar a forma como as aplicações mainframe são atualizadas.

Utilizando a inteligência artificial generativa, a IBM propõe uma ferramenta capaz de traduzir código COBOL em Java, facilitando assim a transição para tecnologias mais modernas.

A iniciativa da IBM

O artigo da InfoWorld apresenta uma solução da IBM, denominada Watsonx Code Assistant for Z, que visa auxiliar clientes do sistema IBM Z na modernização de suas aplicações.

Essa ferramenta, que expande as capacidades do assistente de código baseado em inteligência artificial generativa, é especialmente projetada para traduzir códigos COBOL em Java.

Tal capacidade é crucial, pois o número de desenvolvedores proficientes em COBOL está em declínio, representando um desafio significativo para o desenvolvimento acelerado de aplicações em sistemas mainframe.

A solução é baseada no modelo de código fundacional Watsonx.ai, que permite a geração de novo código Java compatível com o sistema operacional Z.

A ferramenta também é equipada com capacidades de refatoração automática de código, essenciais para extrair serviços empresariais de aplicações monolíticas em COBOL. Essa transformação é feita sem alterar a natureza, o comportamento ou a saída do código original.

Antes de iniciar o processo de refatoração, é necessário utilizar a ferramenta de inteligência de descoberta e entrega de aplicações (ADDI), ainda em desenvolvimento, para compreender as dependências da aplicação alvo e gerar um repositório de metadados.

Após essa etapa, outra ferramenta da IBM é usada para extrair o código COBOL específico que será convertido em Java.

O protótipo atual do Watsonx Code Assistant for Z é capaz de analisar tanto o código COBOL quanto as estruturas de dados dos serviços empresariais extraídos para gerar as classes Java necessárias.

Essa funcionalidade representa um avanço significativo na forma como os

desenvolvedores podem interagir com e modificar o novo código gerado, garantindo que o mesmo seja funcional e eficiente antes de ser implantado.

Reflexões da minha carreira

E pensar que eu iniciei a minha carreira em tecnologia programando na plataforma Mainframe e ouvindo desde então que ela estava prestes a morrer!

Eu lembro como se fosse ontem meu início da carreira na área, de forma totalmente despretensiosa a partir de uma bolsa de estudos em um curso de 2 meses de Cobol, CICS, JCL e DB2, quando eu estava no penúltimo semestre da faculdade.

Na época meus colegas ficaram intrigados com a "loucura" de gastar tempo com uma tecnologia que desde então já estava "morta ou prestes a morrer".

Passados pouco mais de 20 anos, sou muito grato por todas as oportunidades de crescimento na carreira que tive desde esse primeiro passo.

Nunca saberei como teria sido se tivesse seguido algum outro caminho, mas sou bastante feliz com todas as experiências de vida que tive desde então.

Não creio que a plataforma mainframe seja hoje a primeira e muito menos a única opção quando se pensa em iniciar um serviço do zero (o crescimento vertiginoso da cloud – primordialmente em baixa plataforma – mostra isso), mas certamente ainda tem um papel bastante relevante em muitos segmentos.

E ainda deve se manter relevante por um bom tempo.

Há algum tempo comentei que é bem possível que ela siga viva (mesmo que não com a mesma relevância) até mesmo quando chegar a minha aposentadoria em um futuro (espero eu que de longo prazo).

Afinal, Já se foram 20 anos e ela segue com a "morte anunciada", mas ao mesmo tempo com exemplos empíricos que mostram que isso ainda deve levar longos anos.

Mas será que sair da plataforma mainframe é impossível?

Como basicamente tudo em tecnologia, desde que se tenha à disposição tempo e dinheiro, é difícil dizer que algo seja impossível.

Os prós e contras

Li muita coisa sobre a plataforma mainframe z17 trouxe uma série de avanços enquanto interoperabilidade com cloud pública e privada, além do processamento de

AI.

Para muitos casos ela deve seguir sendo adequada, mas ainda assim, sigo igualmente ouvindo que os custos da plataforma mainframe seguem mais altos do que o análogo em outras plataformas (que por sua vez alcançaram alto nível de throughput e disponibilidade)

Daí essa proliferação de casos de migrações do mainframe para outras plataformas.

Creio que essa vantagem financeira tende a ser ainda mais gritante com a ascensão vertiginosa do mundo cloud e da explosão de oferta de provedores e serviços.

E esse novo mercado de cloud está cheio de empresas gigantescas interessadas em conquistar mercado e assim oferecendo deals muito interessantes para contratos de grandes workloads sabidamente "perenes" e recorrentes no longo prazo (que são os casos típicos de quem processa grandes volumes em mainframe).

O equilíbrio de forças do mercado

Como será que se dará o "equilíbrio" do mercado nesse sentido?

Será que a IBM vai baixar seus preços?

Será que ela vai conseguir estancar esses casos de migrações?

Na verdade, qual será a taxa de crescimento ou redução dos clientes mainframe?

Estão realmente reduzindo ou simplesmente crescendo menos que o resto do mercado?

Enfim, gostaria de ter uma visão sobre esses números, apenas por curiosidade mesmo.

Não é um passeio no parque

Mas vale ponderar que trilhar esse caminho de saída não é um passeio no parque.

A dificuldade cresce conforme a complexidade da plataforma/arquitetura de aplicações, especialmente nas grandes empresas com muito processamento, muitas aplicações, muitas integrações e acoplamentos profundos entre si desenvolvidos ao longo de muitos anos (as vezes décadas).

Muitas dessas integrações, acoplamentos e suas regras de negócio possivelmente existem sem o conhecimento de mais nenhuma pessoa ou documentação nas organizações (a vida como ela é) e conforme se avança no processo de migração e vai se descobrindo essa complexidade oculta ou desconhecida, a realidade vai se

descolando do business case original.

Ao mesmo tempo, qualquer um que coloca workloads na cloud percebe rapidamente que os custos de operação podem sair facilmente de controle se não houver um rígido controle e visão crítica.

E por fim, o nível de serviço, que é para muitas indústrias um fator chave.

Afinal, não é uma decisão fácil virar a chave e sair de algo que se sabe que está funcionando e com os serviços de pé e ir para algo que muito provavelmente necessitará de algum fine tuning até operar no mesmo nível de disponibilidade e performance (algo natural em migrações assim).

E essa visão de que na vida real sair do Mainframe não é um passeio no parque é reforçada por diversos cases reais de mercado, como inclusive comentei em um artigo anterior.

O cenário de competição atual

Mas quando olho a evolução recente do universo Cloud cada vez mais maduro, parece que dessa vez o Mainframe finalmente encontrou um concorrente a altura.

Que o digam cases reais de avanços que vão sendo divulgados aqui mesmo no Brasil pelo Santander e Itaú, além do caso da Allianz comentado rm um artigo anterior.

Em paralelo se vê avanços (acho que da própria IBM) em utilizar Gen AI na migração de soluções Cobol para Java. Microsoft e Amazon devem possuir algo análogo.

Os ataques de disrupção estão vindo por todos os lados.

Acho que nunca tantos fatores se uniram em um mesmo momento de forma a promover e facilitar a transformação do Cobol para outras plataformas!

Como comentei há algum tempo, não sei se esse é o “Ragnarok” do Mainframe, mas vale acompanhar os próximos capítulos dessa história nos próximos anos.

A evolução da Plataforma Mainframe

Penso que é de se admirar a incrível flexibilidade e adaptabilidade que a plataforma Mainframe vem demonstrando ao longo de mais de 60 anos de vida.

Uma tecnologia que nasceu em uma época de cartões perfurados, fitas magnéticas, terminais centralizados e processamento em lote continua, ainda hoje, presente no centro de muitas das operações mais críticas do mundo corporativo.

Mais do que isso, ela vem sendo modernizada para dialogar com Cloud, APIs, DevOps, automação, segurança avançada, criptografia pós-quântica e, mais recentemente, Artificial Intelligence.

Também me parece interessante imaginar como seria a competição entre plataformas em uma realidade paralela na qual os custos do mundo Mainframe fossem mais baixos.

Talvez a narrativa dominante sobre substituição, migração e modernização tivesse sido diferente. Talvez muitas empresas tivessem colocado menos energia em tentar retirar workloads críticos dessa plataforma.

Talvez o Mainframe tivesse sido percebido não como um legado caro a ser reduzido, mas como uma plataforma premium, extremamente confiável, altamente especializada e naturalmente adequada para determinados tipos de carga de trabalho.

Ao longo dessas décadas, o Mainframe já enfrentou praticamente todas as grandes ondas tecnológicas: minicomputadores, arquitetura cliente-servidor, Unix, servidores x86, virtualização, web, SOA, Linux, cloud computing, containers, APIs, microsserviços, plataformas digitais e, agora, AI.

Em muitos momentos, foi anunciada a sua obsolescência. Em outros, foi tratado como uma tecnologia do passado. Porém, em vez de desaparecer, foi se reposicionando.

Sua permanência não decorreu de nostalgia tecnológica, mas de uma combinação muito objetiva de fatores: escalabilidade vertical, confiabilidade operacional, segurança, governança, capacidade transacional, compatibilidade histórica e uma engenharia construída para workloads de missão crítica.

A história do Mainframe, portanto, não é apenas a história de uma máquina. É a história de uma arquitetura. É a história de uma forma de pensar computação empresarial. É a história de como algumas plataformas sobrevivem não porque são as mais populares, mas porque resolvem problemas que poucas outras conseguem resolver com o mesmo nível de previsibilidade.

O nascimento de uma plataforma, não apenas de um computador

A origem moderna do Mainframe costuma ser associada ao IBM System/360, anunciado em 7 de abril de 1964. O System/360 foi um divisor de águas porque não representava apenas mais um computador dentro de uma linha de produtos.

Ele representava uma família compatível de sistemas, criada para cobrir diferentes

tamanhos, capacidades e faixas de preço, permitindo que uma organização crescesse sem precisar reescrever completamente seus programas a cada nova evolução de hardware.

A própria IBM descreve o System/360 como o sistema que inaugurou uma nova era de compatibilidade, na qual computadores passaram a ser pensados como plataformas, e não como coleções isoladas de componentes.

Antes desse momento, o mercado de computadores empresariais era marcado por fragmentação. Máquinas diferentes tinham arquiteturas diferentes, instruções diferentes, periféricos diferentes e modelos de programação diferentes.

Uma empresa que adquiria um computador para uma finalidade específica, como contabilidade, processamento científico ou folha de pagamento, frequentemente ficava presa àquele ambiente.

A expansão não era simples. A migração não era fluida. A evolução tecnológica podia significar um redesenho completo das aplicações.

O System/360 atacou diretamente esse problema. A proposta era ambiciosa: criar uma linha de computadores compatíveis entre si, que pudesse atender desde necessidades menores até grandes operações corporativas.

Essa compatibilidade permitia que programas fossem preservados à medida que a empresa evoluía para modelos mais potentes. O impacto foi tão relevante que o System/360 passou a ser lembrado como uma das grandes apostas empresariais da história da tecnologia, frequentemente descrito como um movimento de altíssimo risco para a IBM, mas também como uma das decisões mais transformadoras da indústria.

Esse é um ponto essencial para entender a longevidade do Mainframe. Desde o início, sua proposta não era apenas processar dados.

Sua proposta era proteger investimentos, preservar continuidade e criar uma base tecnológica capaz de evoluir ao longo do tempo.

Em outras palavras, o Mainframe já nasceu com uma preocupação que continua absolutamente atual: compatibilidade evolutiva.

A era em que computação era sinônimo de centralização

Nas décadas de 1960 e 1970, a computação corporativa era naturalmente centralizada. Os computadores eram grandes, caros, complexos e operados por

equipes especializadas. O acesso era feito por cartões perfurados, terminais ou processos em lote.

O usuário final não tinha a relação direta com a tecnologia que passou a existir décadas depois com o computador pessoal, a internet e os smartphones.

Nesse contexto, o Mainframe se consolidou como o grande centro nervoso das corporações.

Bancos, seguradoras, governos, companhias aéreas, empresas de telecomunicações, indústrias e grandes varejistas passaram a depender desses sistemas para processar transações, registrar operações, calcular saldos, emitir faturas, controlar estoques, liquidar pagamentos e manter registros oficiais.

A palavra centralização, muitas vezes vista hoje com certa desconfiança em debates sobre arquitetura, naquele momento era uma virtude.

Centralizar significava controlar. Significava garantir integridade. Significava proteger dados. Significava permitir que grandes volumes de processamento fossem executados com disciplina operacional.

O Mainframe cresceu exatamente nesse ambiente. Ele foi desenhado para ser compartilhado por muitos usuários, muitos programas e muitos processos ao mesmo tempo.

O conceito de multiprogramação, o uso eficiente de recursos computacionais, a priorização de workloads, a administração rigorosa de memória, o controle de acesso e a resiliência operacional passaram a fazer parte de sua identidade técnica.

A computação empresarial nasceu, em grande medida, com essa lógica. Antes da cultura digital contemporânea, antes da computação distribuída e antes da cloud, já existia uma preocupação muito forte com disponibilidade, segurança, performance, eficiência e governança.

O Mainframe foi uma das plataformas que mais incorporou esses princípios.

System/370, virtualização e a maturidade da computação corporativa

Na década de 1970, o System/370 sucedeu o System/360 e aprofundou vários fundamentos que seriam essenciais para a evolução da plataforma. Um dos marcos mais importantes desse período foi o avanço da virtualização.

Embora muitos associem virtualização ao mundo x86 e aos data centers modernos, a

ideia de compartilhar recursos físicos de forma isolada, controlada e eficiente já era explorada há muito tempo no ambiente Mainframe.

Esse ponto costuma ser subestimado. Muitas práticas que posteriormente se tornaram populares na infraestrutura distribuída já eram conhecidas no mundo Mainframe em formas mais maduras e rigorosas.

Isolamento de workloads, particionamento lógico, gestão centralizada de capacidade, alocação dinâmica de recursos e alta utilização do hardware fazem parte da tradição técnica dessa plataforma.

Enquanto o mundo distribuído muitas vezes cresceu por meio da proliferação de servidores, o Mainframe cresceu por meio da consolidação.

Enquanto a infraestrutura aberta, em muitos casos, avançou adicionando mais máquinas, mais instâncias, mais appliances e mais camadas, o Mainframe evoluiu explorando densidade, controle e otimização interna.

Essa diferença moldou duas culturas. De um lado, uma cultura distribuída, orientada à expansão horizontal, à especialização de componentes e à substituição relativamente frequente de tecnologia.

De outro, uma cultura Mainframe, orientada à estabilidade, à continuidade, à máxima utilização de recursos e à preservação de investimentos de longo prazo.

Nenhuma dessas culturas é universalmente superior. Ambas têm virtudes e limitações, mas é importante reconhecer que o Mainframe não sobreviveu por estar parado.

Ele sobreviveu porque evoluiu dentro de uma lógica própria.

O Mainframe como plataforma transacional

A identidade mais forte do Mainframe foi construída em torno do processamento transacional. Em setores como bancos, cartões, seguros, meios de pagamento, companhias aéreas e governo, a transação é a unidade fundamental da operação.

Uma transferência financeira, uma autorização de cartão, uma reserva aérea, uma atualização de saldo, uma liquidação, uma apólice emitida, uma consulta cadastral ou um pagamento processado não são apenas eventos técnicos. São atos de negócio.

Em ambientes desse tipo, alguns atributos deixam de ser desejáveis e passam a ser mandatórios. A transação precisa ser processada corretamente. O dado precisa permanecer íntegro. A operação precisa ser auditável.

A indisponibilidade precisa ser mínima. A segurança precisa ser robusta. A recuperação precisa ser confiável. A consistência precisa prevalecer mesmo sob volume extremo.

É nesse território que o Mainframe encontrou sua maior relevância. A IBM descreve Mainframes como servidores de dados projetados para processar até 1 trilhão de transações web por dia, com altos níveis de segurança e confiabilidade.

A própria definição contemporânea da plataforma continua ligada a essa noção de processamento massivo, crítico e confiável.

O z/Transaction Processing Facility, por exemplo, é descrito pela IBM como um processador voltado a transações de altíssimo volume em ambientes de tempo real.

Esse tipo de posicionamento mostra que o Mainframe permaneceu fortemente associado a contextos nos quais latência, consistência, throughput e disponibilidade não podem ser tratados como variáveis secundárias.

Essa é uma das razões pelas quais muitas iniciativas de migração integral encontraram dificuldades. Migrar uma aplicação simples é uma coisa.

Migrar um ecossistema transacional crítico, altamente integrado, com décadas de regras de negócio, janelas operacionais restritas, dependências regulatórias, integrações batch, interfaces online, auditoria e volumetria extrema é outra completamente diferente.

A chegada da concorrência e o mito da morte do Mainframe

Desde muito cedo, o Mainframe enfrentou concorrência. A própria relevância do System/360 atraiu competidores como Burroughs, UNIVAC, NCR, Control Data e Honeywell, grupo que ficou conhecido historicamente como BUNCH.

Posteriormente, novas ondas tecnológicas prometeram reduzir custos, aumentar flexibilidade e descentralizar o processamento.

Na década de 1980, o computador pessoal mudou a relação dos usuários com a tecnologia. A computação deixou de ser algo distante, fechado no data center, e passou a ocupar mesas, departamentos e áreas de negócio.

A partir daí, o Mainframe começou a ser visto por alguns como símbolo de centralização excessiva.

Na década de 1990, a arquitetura cliente-servidor intensificou esse movimento.

Aplicações passaram a ser distribuídas entre servidores departamentais e estações de trabalho.

Bancos de dados relacionais em plataformas abertas ganharam força. Sistemas Unix e, mais tarde, Windows Server e Linux passaram a disputar workloads corporativos.

A promessa era clara: mais flexibilidade, menor custo aparente, maior autonomia para as áreas e ciclos de desenvolvimento mais rápidos.

Com a internet, essa pressão aumentou. A web trouxe uma nova escala de interação, novos modelos de negócio, novas arquiteturas e uma estética tecnológica muito diferente daquela associada ao Mainframe.

O mundo passou a falar em aplicações web, servidores de aplicação, middleware, portais, e-commerce, integração por serviços e, depois, APIs.

Mais recentemente, cloud computing reposicionou o debate. O argumento deixou de ser apenas técnico e passou a ser também econômico e estratégico.

Elasticidade, consumo sob demanda, automação, infraestrutura como código, serviços gerenciados, disponibilidade global e inovação acelerada tornaram-se atributos centrais das plataformas cloud.

Em todas essas ondas, o Mainframe foi desafiado. Em todas elas, surgiram previsões de que sua relevância estaria chegando ao fim.

Porém, essas previsões ignoraram um ponto importante: workloads diferentes têm naturezas diferentes.

Nem tudo que pode ser executado fora do Mainframe deve necessariamente ser executado fora dele.

Nem tudo que parece mais barato em um cálculo unitário se mostra mais econômico quando são considerados risco, disponibilidade, reengenharia, volume, integração, segurança, compliance, operação e custo total de transformação.

O paradoxo do custo

O custo sempre foi uma das grandes críticas ao Mainframe. Licenciamento de software, MIPS, MSU, consumo de capacidade, contratos de suporte, escassez de profissionais especializados e dependência de fornecedores compõem uma percepção de plataforma cara.

Essa crítica não é irrelevante. Em muitas organizações, o Mainframe representa uma parte expressiva do orçamento de infraestrutura e aplicações.

Mas o debate sobre custo precisa ser tratado com cuidado. O custo de uma plataforma não deve ser avaliado apenas pelo preço unitário de processamento ou armazenamento.

É necessário observar o custo por transação, o custo de indisponibilidade, o custo de risco operacional, o custo de compliance, o custo de migração, o custo de falha, o custo de reescrita de regras de negócio e o custo de perda de conhecimento histórico.

Essa é a razão pela qual tantas iniciativas de substituição acabam sendo mais complexas do que pareciam no business case inicial.

Um workload Mainframe pode parecer caro quando comparado com servidores distribuídos em uma visão simplificada.

Porém, quando se considera a densidade transacional, a disponibilidade, a segurança, a maturidade operacional e a complexidade de recriar décadas de lógica de negócio, o cálculo muda.

Há ainda um aspecto cultural. Muitas organizações passaram anos buscando reduzir o Mainframe por pressão orçamentária, mas sem construir uma compreensão suficientemente granular sobre quais workloads deveriam ser modernizados, quais deveriam ser encapsulados, quais deveriam ser expostos por APIs, quais deveriam ser reescritos e quais deveriam permanecer onde estavam.

Com isso, a modernização muitas vezes foi confundida com substituição.

A modernização mais madura tende a partir de uma premissa diferente. O objetivo não deve ser eliminar uma plataforma por princípio, mas posicionar cada workload no ambiente mais adequado.

Alguns workloads podem ser melhor atendidos em cloud pública. Outros em cloud privada. Outros em plataformas distribuídas tradicionais.

Outros continuam fazendo sentido no Mainframe, especialmente quando criticidade, volume transacional, segurança e continuidade operacional pesam mais do que a moda arquitetural do momento.

Compatibilidade como vantagem estratégica

Uma das características mais impressionantes do Mainframe é a sua compatibilidade histórica. A capacidade de preservar aplicações, dados, linguagens, modelos operacionais e investimentos por décadas é frequentemente vista como sinal de

legado.

Porém, sob outra ótica, essa compatibilidade também pode ser vista como uma forma sofisticada de proteção patrimonial.

Empresas não acumulam apenas código. Elas acumulam conhecimento de negócio. Acumulam regras, exceções, parametrizações, tratamentos regulatórios, integrações, rotinas operacionais e aprendizados que foram incorporados aos sistemas ao longo do tempo.

Em setores regulados, esse conhecimento pode representar uma parte substancial da inteligência operacional da organização.

Quando se fala em modernizar aplicações Mainframe, portanto, não se está falando apenas em converter uma linguagem para outra.

Está se falando em reinterpretar décadas de decisões empresariais. Muitas dessas decisões não estão documentadas. Muitas foram codificadas diretamente em programas COBOL, PL/I, Assembler, JCL, CICS, IMS, DB2 e rotinas batch.

Muitas são conhecidas apenas por profissionais experientes, por operadores, por analistas de sustentação e por usuários que aprenderam a lidar com comportamentos específicos dos sistemas.

Essa é uma das razões pelas quais o Mainframe resiste: não apenas porque o hardware é robusto, mas porque os sistemas que nele vivem carregam conhecimento crítico.

Em muitos casos, o risco de reescrever tudo é maior do que o benefício aparente de sair da plataforma.

A relação com COBOL e o estigma do legado

Não há como contar a história do Mainframe sem mencionar COBOL. A linguagem, criada em uma época muito anterior à internet, continua sendo utilizada em muitos sistemas críticos. Para alguns, isso é sinal de atraso. Para outros, é prova de uma estabilidade notável.

O problema é que o debate público sobre COBOL costuma ser superficial. É comum que se associe a linguagem a sistemas antigos, difíceis de manter e pouco atrativos para novos profissionais.

Parte dessa crítica é verdadeira. Há, sim, escassez de talentos. Há, sim, dificuldades de documentação. Há, sim, desafios de integração com práticas modernas de

desenvolvimento.

Porém, também é verdade que COBOL foi desenhado para processamento de negócios, com legibilidade orientada a regras empresariais, robustez e precisão decimal, atributos relevantes para ambientes financeiros.

A questão central não é demonizar COBOL. A questão é entender como reduzir riscos associados à dependência excessiva de conhecimento antigo, como automatizar testes, como documentar regras, como expor funcionalidades por APIs, como integrar pipelines DevOps, como usar ferramentas de análise estática, como apoiar desenvolvedores com AI e como modernizar a experiência de desenvolvimento sem necessariamente jogar fora sistemas que funcionam.

Nesse sentido, a modernização Mainframe não precisa ser uma decisão binária entre manter tudo como está ou reescrever tudo do zero.

Entre esses extremos, há um amplo conjunto de alternativas: refatoração seletiva, replatforming, encapsulamento, exposição por serviços, extração de dados, decomposição de domínios, integração com eventos, automação de testes, modernização de interfaces, observabilidade e uso de AI para compreensão de código.

Linux no Mainframe e a abertura da plataforma

Um dos momentos mais relevantes da reinvenção do Mainframe foi a adoção de Linux na plataforma. A chegada do Linux ao Mainframe ajudou a mudar a percepção de que se tratava de um ambiente fechado e isolado.

A possibilidade de executar workloads Linux em uma plataforma com características de Mainframe criou um caminho interessante para consolidação, virtualização e integração com ecossistemas abertos.

Em 2002, por exemplo, a IBM já anunciava iniciativas relacionadas a Mainframes com Linux, em uma tentativa de tornar a plataforma mais acessível e conectada a novos perfis de workloads.

Esse movimento foi importante porque mostrou que o Mainframe poderia abrigar não apenas aplicações tradicionais, mas também componentes modernos baseados em sistemas abertos.

A adoção de Linux também reforçou uma característica que muitos esquecem: o Mainframe é, antes de tudo, uma plataforma de execução. Ele não é apenas z/OS. Ele

pode suportar diferentes ambientes e formas de processamento.

Essa flexibilidade permitiu que empresas explorassem consolidação de servidores, workloads open source, aplicações Java, bancos de dados, middleware e integrações modernas em um ambiente com altíssima disponibilidade e forte isolamento.

Esse foi um ponto de inflexão na narrativa. O Mainframe deixou de ser apenas o lugar onde sistemas antigos continuavam rodando e passou a poder ser visto também como uma plataforma capaz de hospedar workloads novos, especialmente quando densidade, segurança e governança eram relevantes.

Do e-business à API Economy

Com a popularização da internet e, posteriormente, da API Economy, o Mainframe precisou se conectar a um mundo cada vez mais distribuído.

Durante muito tempo, a integração entre sistemas centrais e canais digitais foi feita por camadas intermediárias, barramentos, filas, mensageria, ESBs, adaptadores e serviços.

Essa camada de integração foi fundamental para permitir que bancos, seguradoras, companhias aéreas e governos expusessem funcionalidades críticas para internet banking, mobile banking, portais, aplicativos, parceiros, fintechs, marketplaces e ecossistemas digitais.

Em muitos casos, o usuário final interagia com uma interface moderna, mas a transação final continuava sendo registrada ou autorizada em um sistema Mainframe.

Esse padrão continua relevante. Uma arquitetura digital moderna raramente é composta apenas por sistemas cloud-native.

Ela costuma ser um arranjo híbrido, no qual canais digitais, microsserviços, plataformas de dados, sistemas de integração, motores de decisão, soluções de segurança e sistemas centrais convivem.

O Mainframe, nesse contexto, não precisa ser invisível nem isolado. Ele precisa ser integrado com disciplina arquitetural.

A exposição de capacidades Mainframe por APIs foi uma das formas mais pragmáticas de modernização.

Em vez de remover imediatamente o core, muitas empresas optaram por encapsular funcionalidades, proteger a integridade transacional e acelerar a entrega digital por meio de camadas modernas de integração.

Essa abordagem permite preservar o que é crítico, ao mesmo tempo em que se melhora a experiência do cliente, a velocidade de desenvolvimento e a capacidade de integração com parceiros.

Mainframe e Cloud: competição, convivência e complementaridade

A relação entre Mainframe e Cloud é frequentemente apresentada como competição. De um lado, uma plataforma centralizada, de alto custo, com herança histórica e forte presença em grandes empresas.

De outro, uma plataforma elástica, distribuída, moderna, com consumo sob demanda e ritmo acelerado de inovação.

Essa oposição é compreensível, mas incompleta. Em muitas organizações, a realidade não é Mainframe ou Cloud. A realidade é Mainframe e Cloud.

O ambiente corporativo moderno é híbrido por natureza. Aplicações legadas, SaaS, cloud pública, cloud privada, data centers próprios, edge computing, plataformas de dados, APIs, containers e sistemas centrais coexistem.

A pergunta estratégica correta não deveria ser apenas: como sair do Mainframe? A pergunta mais madura deveria ser: qual é o papel adequado do Mainframe no target architecture da organização?

Em alguns casos, workloads podem e devem migrar. Em outros, a melhor decisão é modernizar no lugar.

Em outros, pode ser interessante desacoplar partes da lógica, mover dados para plataformas analíticas, expor serviços, reduzir dependências, automatizar operação e melhorar a governança de custos.

Em outros ainda, o Mainframe pode continuar sendo o sistema de registro transacional, enquanto a inovação de experiência, analytics e canais digitais ocorre em cloud.

A própria IBM posiciona o z17 como uma plataforma conectada a AI e hybrid cloud, ressaltando integração com dados e aplicações críticas.

Isso indica que a estratégia atual não é tentar proteger o Mainframe por isolamento, mas inseri-lo em uma arquitetura híbrida e moderna.

Segurança, resiliência e o valor do controle

A segurança sempre foi um dos atributos mais relevantes do Mainframe. Em setores regulados, não basta que uma plataforma seja funcional.

Ela precisa ser controlável, auditável, segregável, resiliente e compatível com exigências de compliance.

A proteção de dados críticos, a rastreabilidade de acessos, a criptografia, o gerenciamento de identidades, a segregação de funções e a capacidade de recuperação são partes integrantes desse contexto.

A IBM posiciona o IBM Z com foco em segurança avançada, incluindo casos de uso como detecção de fraudes em tempo real no setor financeiro.

Também destaca recursos de resiliência cibernética, como o IBM Z Cyber Vault, voltado à proteção de dados críticos e recuperação após ataques, incluindo ameaças como ransomware.

Esse ponto é cada vez mais importante. A discussão contemporânea sobre resiliência não se limita à disponibilidade tradicional. Não basta manter o sistema no ar.

É necessário preservar integridade, detectar manipulação indevida de dados, recuperar ambientes comprometidos, resistir a ataques cibernéticos e demonstrar conformidade regulatória.

Nesse sentido, o Mainframe tem uma vantagem cultural. Ele nasceu em ambientes nos quais controle, segregação, auditoria e operação disciplinada eram mandatórios.

Enquanto muitas plataformas modernas estão reforçando agora práticas de governança que cresceram em importância com regulação, risco cibernético e soberania de dados, o Mainframe já traz essa mentalidade em seu desenho operacional.

Isso não significa que o Mainframe seja automaticamente seguro. Nenhuma plataforma é.

Configurações inadequadas, acessos excessivos, falhas de governança, ausência de atualização, dependência de conhecimento antigo e integrações mal desenhadas podem criar riscos relevantes. Mas significa que a plataforma oferece fundamentos robustos para ambientes que exigem controle rigoroso.

A era do zSeries, IBM Z e a continuidade da evolução

Ao longo do tempo, a nomenclatura e as gerações da plataforma evoluíram. System/360, System/370, ESA, S/390, zSeries, System z e IBM Z representam uma trajetória de continuidade e renovação.

Cada geração incorporou avanços de processamento, memória, I/O, virtualização, segurança, criptografia, disponibilidade e integração.

A letra z passou a ser associada à ideia de zero downtime, reforçando a proposta de alta disponibilidade.

Embora nenhuma tecnologia deva ser tratada como infalível, essa ambição comunica bem o posicionamento da plataforma: suportar operações que não podem parar.

A modernização do IBM Z também acompanhou mudanças no tipo de workload. Além das cargas tradicionais em COBOL, CICS, IMS e DB2, a plataforma passou a dialogar com Java, Linux, containers, APIs, DevOps, observabilidade, automação, criptografia avançada e AI.

O objetivo deixou de ser apenas manter sistemas antigos e passou a incluir a modernização contínua de aplicações críticas.

Essa evolução é relevante porque desmonta uma visão simplista do Mainframe como tecnologia congelada. O que se mantém é a continuidade arquitetural.

O que muda são as formas de integração, os recursos de hardware, os modelos operacionais e as possibilidades de desenvolvimento.

AI no Mainframe: a próxima reinvenção

A fase mais recente dessa trajetória é a aproximação entre Mainframe e Artificial Intelligence. Essa aproximação não deve ser vista apenas como marketing tecnológico.

Há uma lógica prática por trás dela. Muitos dos dados mais críticos das organizações continuam nos sistemas centrais.

Em bancos, seguradoras, governos e grandes empresas, transações de alto valor, registros oficiais, históricos de clientes, saldos, limites, contratos e eventos operacionais continuam fortemente conectados ao Mainframe.

A aplicação de AI próxima desses dados pode reduzir movimentos desnecessários,

diminuir latência, preservar segurança e permitir decisões em tempo real.

A IBM afirma que AI on IBM Z permite aplicar machine learning diretamente aos dados transacionais, evitando movimentação de dados e habilitando insights em tempo real.

Esse posicionamento é particularmente relevante para casos como detecção de fraude, análise de risco, classificação de transações, monitoramento de comportamento, prevenção de anomalias, automação operacional e assistência ao desenvolvimento.

Em ambientes financeiros, por exemplo, a capacidade de avaliar uma transação no momento em que ela ocorre pode ser muito mais valiosa do que analisá-la posteriormente em uma plataforma analítica separada.

O IBM z17, anunciado em 2025, reforça essa direção. A IBM o descreveu como um Mainframe totalmente projetado para a era da AI, com disponibilidade geral prevista para 18 de junho de 2025 e com o IBM Spyre Accelerator esperado a partir do quarto trimestre de 2025.

A página do produto destaca aceleração de AI, uso de modelos múltiplos, insights em tempo real, segurança aprimorada, produtividade e resiliência.

A questão aqui não é imaginar que todo processamento de AI será feito no Mainframe. Isso não faria sentido.

Modelos massivos, treinamento em larga escala, experimentação científica e workloads intensivos de GPU continuarão sendo executados em ambientes especializados.

O ponto é outro: certas inferências, decisões e automações podem fazer sentido perto dos dados transacionais e dos sistemas de registro. Nessa fronteira, o Mainframe pode ter um papel importante.

Mainframe como plataforma de confiança

Ao se observar a história completa, parece claro que o principal ativo do Mainframe não é apenas performance. Também não é apenas compatibilidade. O principal ativo é confiança operacional.

Confiança, nesse contexto, significa que a plataforma foi testada durante décadas em ambientes de altíssima criticidade.

Significa que processos de negócio inteiros foram construídos ao seu redor. Significa que equipes operacionais desenvolveram práticas, controles e rotinas altamente

especializadas.

Significa que reguladores, auditores e executivos aprenderam a confiar na estabilidade desses sistemas.

Essa confiança tem valor econômico.

Uma plataforma que processa transações críticas com previsibilidade reduz riscos.

Uma plataforma que mantém disponibilidade elevada protege receita.

Uma plataforma que preserva integridade de dados evita perdas financeiras, impactos reputacionais e sanções regulatórias.

Uma plataforma que permite rastreabilidade facilita auditorias.

Uma plataforma que já contém décadas de regras de negócio reduz incerteza funcional.

O problema é que confiança operacional nem sempre aparece bem em apresentações comparativas de custo.

Ela não é facilmente capturada em um gráfico de preço por core, preço por gigabyte ou preço por instância. Mas aparece rapidamente quando há uma falha grave, uma migração mal-sucedida, uma indisponibilidade sistêmica ou uma inconsistência de dados.

É por isso que o Mainframe continua presente nos setores em que o erro custa caro.

O desafio geracional e a escassez de talentos

Apesar de sua força técnica, a plataforma enfrenta um desafio real: pessoas. Muitos profissionais experientes de Mainframe estão se aposentando ou se aproximando dessa fase.

Ao mesmo tempo, novas gerações de desenvolvedores tendem a se formar em linguagens, ferramentas e ambientes mais associados à web, cloud, mobile, dados e AI.

Esse desalinhamento cria riscos. A escassez de profissionais pode aumentar custos, dificultar manutenção, reduzir velocidade de evolução e concentrar conhecimento em poucos especialistas.

Esse talvez seja um dos maiores riscos estratégicos para empresas dependentes de Mainframe.

A resposta não deve ser apenas contratar mais especialistas, até porque o mercado pode não oferecer volume suficiente.

Deve ser construída uma estratégia de gestão de conhecimento, renovação de talentos, documentação assistida por ferramentas, automação de testes, modernização de ambientes de desenvolvimento, integração com práticas DevOps e uso de AI para apoiar compreensão de código legado.

Ferramentas modernas podem ajudar novos profissionais a interagir com sistemas antigos de forma mais produtiva.

Ambientes de desenvolvimento integrados, pipelines automatizados, análise estática de código, geração de documentação, testes automatizados e assistentes baseados em AI podem reduzir a barreira de entrada.

O futuro do Mainframe, portanto, dependerá menos da capacidade de preservar a cultura antiga e mais da capacidade de traduzi-la para uma nova geração de engenharia.

Modernizar não é necessariamente migrar

Um dos maiores equívocos no debate sobre Mainframe é tratar modernização como sinônimo de migração.

Em muitas organizações, essa associação criou programas caros, longos e arriscados, movidos mais por pressão tecnológica do que por uma análise clara de valor.

Modernizar pode significar muitas coisas. Pode significar melhorar a experiência do desenvolvedor. Pode significar criar APIs para funcionalidades centrais. Pode significar automatizar testes. Pode significar reduzir dependências batch. Pode significar separar dados analíticos de dados transacionais. Pode significar reescrever módulos específicos. Pode significar mover partes menos críticas para cloud. Pode significar melhorar observabilidade. Pode significar otimizar consumo de capacidade. Pode significar revisar contratos. Pode significar atualizar práticas de segurança. Pode significar documentar regras de negócio.

A migração é apenas uma das alternativas. Em alguns casos, ela será correta. Em outros, será perigosa. Em muitos, será economicamente questionável. A decisão deve ser baseada em critérios arquiteturais e de negócio, não em preferência ideológica por uma plataforma.

Uma abordagem madura deveria classificar o portfólio Mainframe em grupos. Sistemas com alta criticidade e alto acoplamento talvez devam ser modernizados no

lugar.

Componentes com baixa criticidade e alto custo talvez devam ser migrados. Regras de negócio reutilizáveis talvez devam ser expostas por APIs. Dados necessários para analytics talvez devam ser replicados de forma governada.

Processos batch talvez devam ser redesenhados. Interfaces antigas talvez devam ser substituídas. O core transacional talvez deva permanecer.

Essa visão evita tanto o conservadorismo excessivo quanto a modernização imprudente.

A arquitetura híbrida como destino natural

O destino mais provável para o Mainframe não é o desaparecimento imediato, mas a integração cada vez mais sofisticada em arquiteturas híbridas.

As empresas continuarão usando cloud para inovação, elasticidade, analytics, AI, colaboração, SaaS e canais digitais. Também continuarão usando sistemas centrais para registros críticos, processamento transacional e operações reguladas.

A arquitetura vencedora será aquela capaz de orquestrar esse conjunto com clareza. Isso exige governança de APIs, gestão de dados, observabilidade ponta a ponta, segurança integrada, identidade federada, automação, controle de custos, resiliência e arquitetura empresarial.

Nesse cenário, o Mainframe deve deixar de ser tratado como um bloco isolado dentro do data center.

Ele deve ser tratado como um asset tecnológico estratégico, com funções claras dentro do operating model de tecnologia. Essa mudança de postura é importante.

Quando uma plataforma é tratada apenas como legado, tende a ser negligenciada. Quando é tratada como asset crítico, tende a ser governada, modernizada e otimizada.

O desafio das organizações não é amar ou odiar o Mainframe. O desafio é saber o que fazer com ele.

O que a história do Mainframe ensina

sobre tecnologia

A história do Mainframe oferece algumas lições importantes para qualquer executivo de tecnologia.

A primeira lição é que longevidade não significa obsolescência. Algumas tecnologias permanecem porque foram bem desenhadas, porque resolvem problemas fundamentais e porque continuam evoluindo. O tempo, por si só, não torna uma plataforma irrelevante.

A segunda lição é que compatibilidade tem valor. Em um mercado frequentemente obcecado por rupturas, é fácil esquecer que empresas precisam preservar investimentos, reduzir riscos e garantir continuidade. A inovação precisa conviver com a responsabilidade operacional.

A terceira lição é que arquitetura importa. O Mainframe não sobreviveu apenas por força comercial. Ele sobreviveu porque sua arquitetura foi adequada a certos problemas críticos. Escalabilidade, segurança, disponibilidade, isolamento e integridade são características difíceis de improvisar depois.

A quarta lição é que custo precisa ser analisado em múltiplas dimensões. O mais barato em infraestrutura pode não ser o mais barato em risco. O mais barato em licenciamento pode não ser o mais barato em reengenharia. O mais barato no curto prazo pode não ser o mais econômico no ciclo completo.

A quinta lição é que modernização inteligente é seletiva. Nem tudo precisa ser reescrito. Nem tudo deve ser preservado. Nem tudo deve ir para cloud. Nem tudo deve ficar no Mainframe. A boa arquitetura é aquela que posiciona cada capability tecnológica no lugar mais adequado.

Uma plataforma que atravessou eras

Quando se olha para a trajetória completa, impressiona perceber que o Mainframe atravessou praticamente todas as eras da computação empresarial.

Ele nasceu na era da computação centralizada. Adaptou-se à era transacional. Sobreviveu à ascensão dos minicomputadores. Resistiu ao computador pessoal. Conviviu com cliente-servidor. Integrado à web, sustentou internet banking, reservas, pagamentos e grandes sistemas governamentais. Abriu-se ao Linux. Conectou-se por APIs. Passou a fazer parte da arquitetura híbrida. Agora, busca se posicionar também na era da AI.

Poucas plataformas podem reivindicar uma história semelhante. A maioria das tecnologias corporativas tem ciclos muito mais curtos. Linguagens, frameworks, bancos de dados, servidores de aplicação, appliances, ferramentas de integração e modelos arquiteturais surgem, crescem, amadurecem e declinam em poucas décadas, às vezes em poucos anos.

O Mainframe, por sua vez, continua sendo parte da espinha dorsal de muitas organizações críticas.

Essa permanência não deve ser romantizada. A plataforma tem desafios reais. Custo, talentos, complexidade, dependência de fornecedores, percepção de legado, dificuldade de mudança e integração com modelos modernos são temas legítimos. Porém, também não deve ser subestimada. Sua capacidade de reinvenção é rara.

A hipótese dos custos mais baixos

É interessante voltar à provocação inicial: o que teria acontecido se os custos do Mainframe fossem mais baixos?

Provavelmente, a pressão por substituição teria sido menor. Muitos debates arquiteturais teriam sido conduzidos de forma menos emocional e mais técnica.

O Mainframe talvez tivesse sido mais amplamente adotado para workloads que exigem alta densidade, resiliência e segurança.

Algumas empresas talvez tivessem evitado arquiteturas distribuídas excessivamente complexas, criadas para escapar de custos centrais, mas que depois geraram novos custos de integração, observabilidade, segurança e operação.

Por outro lado, custos mais baixos também poderiam ter reduzido a pressão por inovação externa. Parte da evolução tecnológica ocorreu justamente porque empresas buscaram alternativas mais flexíveis e acessíveis.

A concorrência de plataformas abertas, cloud e arquiteturas distribuídas forçou o Mainframe a se modernizar. Nesse sentido, a pressão competitiva foi saudável.

Talvez a conclusão mais equilibrada seja reconhecer que o Mainframe foi, ao mesmo tempo, vítima e beneficiário de sua própria proposta de valor.

Seu custo elevado limitou sua expansão e alimentou projetos de substituição. Mas sua robustez, confiabilidade e capacidade transacional sustentaram sua permanência. A plataforma não venceu por ser barata. Venceu, nos contextos em que continuou relevante, por ser difícil de substituir.

O futuro provável

O futuro do Mainframe não deve ser lido como uma repetição simples do passado. A plataforma continuará relevante, mas seu papel será mais seletivo.

A tendência não parece ser um retorno ao mundo totalmente centralizado. Tampouco parece realista imaginar uma eliminação rápida dos Mainframes em grandes organizações reguladas.

O mais provável é uma convivência. Mainframes continuarão sustentando sistemas críticos, especialmente em setores que dependem de processamento transacional massivo, segurança, integridade e resiliência.

Ao mesmo tempo, esses ambientes serão cada vez mais integrados a cloud, plataformas digitais, AI, analytics e automação.

A modernização será menos sobre trocar uma plataforma por outra e mais sobre reduzir acoplamentos, aumentar interoperabilidade, melhorar governança de dados, acelerar desenvolvimento, renovar talentos e aplicar AI de forma pragmática sobre sistemas e dados críticos.

O Mainframe do futuro talvez seja menos visível para o usuário final, mas continuará essencial em determinadas cadeias de valor. Ele será uma plataforma de bastidor, mas um bastidor crítico. Uma infraestrutura de confiança. Um motor transacional. Um repositório de conhecimento histórico. Um componente da arquitetura híbrida empresarial.

Um legado que ainda está em movimento

A palavra legado costuma carregar uma conotação negativa em tecnologia. Muitas vezes, legado é usado como sinônimo de antigo, rígido, caro e problemático.

Mas legado também pode significar algo que foi herdado porque teve valor. Algo que permanece porque sustentou a organização. Algo que precisa ser cuidado, modernizado e governado, não simplesmente descartado.

O Mainframe é talvez um dos melhores exemplos dessa ambiguidade. Ele é legado no sentido histórico. Mas não é apenas legado no sentido depreciativo. É uma plataforma viva, que continua sendo atualizada, reposicionada e integrada a novas ondas tecnológicas.

Sua história mostra que a tecnologia corporativa não evolui apenas por substituição. Muitas vezes, ela evolui por camadas. O novo não elimina imediatamente o antigo.

O digital não apaga o transacional. A cloud não elimina automaticamente o data center. AI não substitui a necessidade de dados confiáveis. A modernização não elimina a importância da continuidade.

Ao final, talvez o Mainframe continue existindo porque as empresas continuam precisando daquilo que ele sempre prometeu entregar: confiança em escala.

E, em um mundo cada vez mais digital, volátil, regulado, integrado e exposto a riscos cibernéticos, confiança em escala continua sendo uma das capacidades tecnológicas mais valiosas que uma organização pode possuir.

Concluindo

Refletir sobre minha jornada desde os primórdios da minha carreira em tecnologia, onde comecei programando em plataformas Mainframe, até o presente, é uma experiência que evoca tanto nostalgia quanto admiração pelo avanço da tecnologia.

Durante essas décadas, sempre ouvi rumores sobre o suposto ocaso do Mainframe.

No entanto, a resiliência dessa tecnologia tem sido notável, desafiando repetidamente as previsões de sua obsolescência.

Iniciei minha carreira de maneira despretensiosa, aprendendo Cobol, CICS, JCL e DB2 em um curso de dois meses financiado por uma bolsa de estudos.

Naquela época, muitos colegas questionavam a lógica de investir tempo em uma tecnologia considerada por muitos como ultrapassada.

Olhando para trás, após mais de vinte anos, sou profundamente grato pelas oportunidades que essa escolha inicial me proporcionou.

Essa trajetória não apenas moldou minha carreira profissional, mas também enriqueceu minha vida com experiências inestimáveis.

A plataforma Mainframe, embora não seja a primeira opção para projetos novos — especialmente com o crescimento exponencial da computação em nuvem —, continua desempenhando um papel crucial em diversos setores.

Sua capacidade de processar grandes volumes de dados e sua confiabilidade em operações críticas ainda são comparativamente superiores em muitos aspectos.

A iniciativa da IBM em desenvolver o Watsonx Code Assistant for Z exemplifica um passo significativo em direção à modernização de sistemas mainframe, que são cruciais para muitas operações empresariais, mas frequentemente limitados por tecnologias desatualizadas. Essa solução não apenas facilita a transição de COBOL

para Java, uma linguagem mais moderna e amplamente usada, mas também otimiza o processo ao minimizar os riscos associados à alteração de código funcional.

Em minha opinião, a implementação dessa tecnologia pela IBM reflete uma visão estratégica notável, que não apenas aborda uma necessidade imediata de mercado, mas também demonstra o potencial das inteligências artificiais generativas no campo do desenvolvimento de software.

Este desenvolvimento não apenas reduzirá significativamente os custos e o tempo necessários para a modernização de sistemas legados, mas também permitirá que as empresas aproveitem melhor suas infraestruturas existentes enquanto se adaptam às novas demandas tecnológicas.

A abordagem da IBM reforça a importância de investimentos contínuos em inovação e desenvolvimento tecnológico, garantindo que as ferramentas de IA, como o Watsonx Code Assistant for Z, sejam não apenas funcionalidades adicionais, mas sim elementos transformadores na jornada de modernização digital.

Adicionalmente, a evolução contínua do Mainframe, que agora inclui capacidades modernas como integração com a inteligência artificial e a computação em nuvem, sinaliza que essa tecnologia ainda pode se adaptar e permanecer relevante em um mercado em constante transformação.

Enquanto observamos o desenvolvimento da competição entre plataformas tradicionais e soluções em nuvem, a decisão de migrar ou manter operações em Mainframes deve ser baseada em uma análise cuidadosa das necessidades específicas e capacidades organizacionais.

Finalmente, como líder e profissional de TI que acompanhou e liderou transformações significativas, afirmo que a adaptação e inovação são fundamentais para a sobrevivência e o sucesso em um ambiente empresarial que evolui rapidamente.

Neste processo, preservar o conhecimento e integrar novas tecnologias de forma estratégica são essenciais para garantir a competitividade e a eficácia operacional no futuro.