



Curso Computação em Nuvem
Engenharia de computação

Projeto da Disciplina

Tema:

Plataforma Escalável em Nuvem para Conversão de Imagens em Base64

Segundo Semestre de 2025

Versão	Elaboração/Revisão	Data	Aprovação
1	Tiago Demay / João Luisi	26/08/2025	

História e Contexto

A empresa **Demay's Infra Company**, especializada em soluções de infraestrutura em nuvem, foi procurada por um cliente que deseja oferecer um serviço moderno de conversão de imagens. O desafio consiste em desenvolver uma plataforma completa, aproveitando serviços da **AWS**, que permita aos usuários finais enviarem imagens, convertê-las para **Base64**, armazená-las em um banco de dados gerenciado e recuperá-las quando necessário. Para atender a essa demanda, os grupos de alunos atuarão como a *equipe de projeto* da Demay's Infra Company, sendo responsáveis por todo o ciclo de concepção, implantação e documentação da solução.

Além do serviço principal, a empresa para a qual a Demay's Infra Company está trabalhando **não aceita correr riscos desnecessários de indisponibilidade** ou aprisionamento tecnológico. Por isso, este projeto inclui, de forma **obrigatória e transversal**, a elaboração de um **Plano Oficial de Disaster Recovery (DR) / Cloud Exit**, descrevendo como a solução seria migrada da AWS para outra nuvem (por exemplo, *Azure* ou *Google Cloud*) em caso de falha prolongada, exigência regulatória ou decisão estratégica.

O Plano de DR/Cloud Exit será avaliado por **rubrica específica**, e seu nível **Regular (2)** em todos os critérios é requisito mínimo para aprovação do projeto. Dessa forma, a solução final entrega não apenas funcionalidade e escalabilidade em nuvem, mas também **resiliência operacional** e **capacidade de migração** em caso de necessidade.

Descrição do Projeto

O sistema é composto por:

- Site Estático Institucional: hospedado em bucket S3 com páginas em HTML/CSS/JS. O site apresenta:
 1. História da empresa fictícia.
 2. Uma aba com informações sobre o grupo desenvolvedor do projeto.
 3. Uma aba de acesso à área de clientes.
- Área de Clientes: hospedada em instâncias EC2, atrás de um Load Balancer (ALB) e um Auto Scaling Group (ASG), garantindo alta disponibilidade e escalabilidade. Nessa área o cliente pode:
 1. Fazer upload de uma imagem.
 2. Listar as imagens armazenadas (identificadas por *tags*).
 3. Visualizar a conversão da imagem em formato Base64.
- Documento de DR:
 1. Definir cenários de falha considerados;
 2. Mapear serviços equivalentes entre AWS e a nuvem de destino (escolhida);
 3. Apresentar procedimentos de preparação, ativação (em crise), validação, rollback e comunicação;
 4. Estimar custos financeiros e operacionais da transição (egresso de dados, recursos no destino, horas-equipe) e o cronograma de execução;

Arquitetura Proposta

A arquitetura utiliza serviços gerenciados da AWS:

- **Amazon S3:** hospeda o site institucional e recebe os uploads de imagens dos clientes.
- **AWS Lambda:** é acionado automaticamente a cada novo upload no S3. O Lambda converte a imagem em Base64 e grava os dados no DynamoDB.
- **Amazon DynamoDB:** armazena as informações das imagens convertidas, incluindo metadados (content-type, tamanho, data de criação) e os dados em Base64 (armazenados em *chunks* caso a imagem seja grande).
- **Amazon EC2:** executa a aplicação da área de clientes (API e front-end dinâmico).
- **Application Load Balancer (ALB):** expõe a área de clientes via HTTPS, distribui requisições entre instâncias EC2.
- **Auto Scaling Group (ASG):** aumenta ou diminui a quantidade de instâncias EC2 conforme a demanda.
- **VPC e Security Groups:** garantem isolamento da rede e regras de segurança mínimas, como acesso externo apenas via ALB e comunicação interna controlada.

Fluxo do Sistema

1. O cliente acessa o site institucional (S3) e navega até a *Área de Clientes*.
2. Na área de clientes (EC2), solicita uma URL pré-assinada para upload da imagem.
3. A imagem é enviada diretamente ao S3.
4. O evento *ObjectCreated* aciona uma função Lambda.
5. O Lambda lê a imagem do S3, converte em Base64, quebra em *chunks* se necessário, e armazena no DynamoDB junto com os metadados.
6. Quando o cliente requisita uma imagem, a aplicação no EC2 consulta o DynamoDB, recompõe a string Base64 e a retorna ao cliente.
7. O cliente visualiza a string Base64 no console do navegador ou em uma tag `` embutida.

Observação do item 2. - URL Pré-assinada

Uma **URL pré-assinada (pre-signed URL)** é um link temporário, gerado pelo back-end (no nosso caso, a aplicação da *Área de Clientes* rodando em EC2), que concede permissão limitada para operações específicas em um objeto do Amazon S3 **sem expor** credenciais da AWS ao usuário final.

Por que usar?

- **Segurança:** o cliente não recebe chaves IAM. A autorização é embutida na URL, com *deadline* curto (ex.: 5 minutos) e escopo restrito (método HTTP, nome do arquivo, content-type).
- **Eficiência:** o arquivo é enviado diretamente do navegador para o S3, sem “passar” por EC2, reduzindo tráfego, latência e custo.
- **Controle:** o servidor define o bucket, o caminho (*key*), os tipos aceitos (ex.: image/png, image/jpeg) e o tempo de validade.

Resumo do Fluxo

1. O cliente (navegador) solicita à API da *Área de Clientes* (EC2) uma URL pré-assinada para subir foto.png.
2. A API (EC2) gera a URL pré-assinada usando suas credenciais IAM e retorna a URL ao navegador.
3. O navegador envia a imagem diretamente ao S3 usando a URL pré-assinada (método PUT).

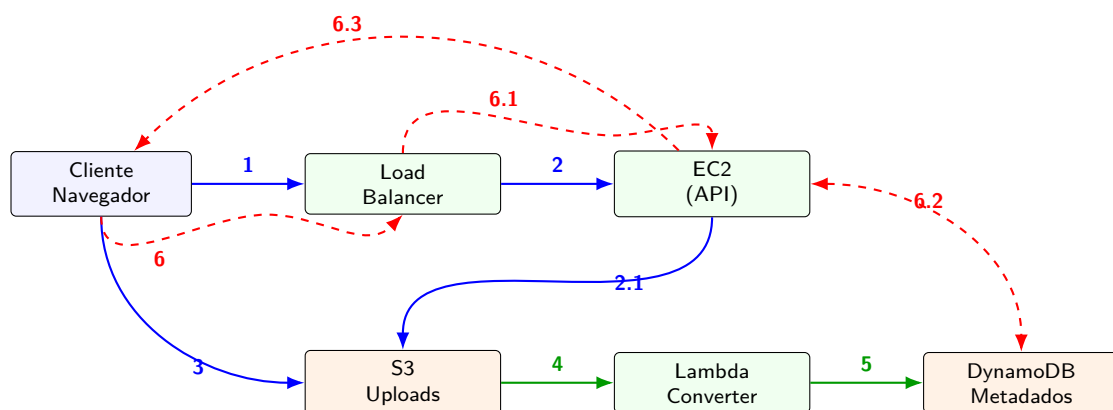


Figura 1: Fluxo de upload com URL pré-assinada (S3) e processamento assíncrono

Legenda:

Upload (1-3): 1-Solicita URL, 2-Gera URL, 2.1-Assina, 3-Upload direto S3

Processamento (4-5): 4-Evento S3, 5-Converte para Base64

Leitura (6): 6-Requisita, 6.1-API, 6.2-Consulta DB, 6.3-Retorna

Requisitos Funcionais

- RF1: O site institucional deve ser acessível via bucket S3 (com páginas: Empresa, Grupo, Área de Clientes).
- RF2: A área de clientes deve estar disponível via ALB, com instâncias EC2 em ASG.
- RF3: O upload de imagens deve ocorrer via URL pré-assinada gerada pela aplicação no EC2.

- RF4: O Lambda deve converter a imagem para Base64 e salvar os dados no DynamoDB.
- RF5: O cliente deve ser capaz de listar suas imagens armazenadas e visualizar a string Base64 de uma imagem escolhida.

Requisitos Não Funcionais

- RNF1: A solução deve ser escalável e tolerante a falhas.
- RNF2: Devem ser aplicadas boas práticas de segurança (least privilege em IAM, SGs restritos, buckets privados).
- RNF3: A arquitetura deve ser *cost-aware*, utilizando instâncias de baixo custo e DynamoDB sob demanda.

Rubrica de Avaliação (Infraestrutura 50% da nota)

Estágio / Critério	Excelente (4)	Bom (3)	Regular (2)	Insuficiente (1)
1 – Estrutura mínima	Infra completa (S3, VPC, SGs, EC2) + documentação clara	Infra criada com documentação parcial	S3 + EC2 básicos com prints mínimos	Sem entrega ou sem documentação
2 – Upload e Conversão	Fluxo completo, com metadados e chunking, bem documentado	Upload + Base64 funcionam, documentação incompleta	Upload dispara Lambda e salva em DynamoDB, doc básica	Sem fluxo ou documentação
3 – Área de Clientes + Escalabilidade	Área funcional (ALB + ASG + HTTPS), listagem e testes, doc clara	Área funcional com ALB/ASG, doc sem evidência de escalabilidade	Área acessível e listagem básica, doc simples	Área não entregue ou sem documentação
4 – Projeto Avançado (Segurança, SLA e Custo)	Relatório completo, diagrama, SLA, custo, segurança aplicada	Relatório bom, sem SLA/custo detalhados	Relatório básico em alto nível, sem SLA/custo	Relatório ausente ou incoerente

Rubrica de Avaliação – Plano de DR 50% da nota

Critério (DR)	Excelente (4)	Bom (3)	Regular (2)	Insuficiente (1)
Objetivos e Estratégia (RTO/RPO & cenários)	RTO/RPO por serviço; cenários mapeados; estratégia (cold/warm/active) bem justificada	RTO/RPO definidos; principais cenários; justificativa parcial	RTO/RPO em alto nível; 1 cenário; sem trade-offs	Sem RTO/RPO ou estratégia
Mapeamento de Serviços & Portabilidade	AWS→Azure/GCP completo (S3, Lambda, DynamoDB, EC2/ASG, ALB, IAM, DNS, certs); riscos/limites	Mapeamento majoritário; pontos críticos descritos	Mínimo (storage, compute, função) sem detalhes	Sem mapeamento
Procedimentos e Runbooks	Passo a passo claro; pré-reqs; validações; rollback; comunicação	Passos claros; validações básicas	Alto nível; sem validação/rollback	Incompleto/ambíguo
Rede, Segurança e Identidade	VPC→VNet/VPC; SG→NSG/Firewall; IAM→RBAC; TLS/certs; segredos	Itens principais mapeados; poucas lacunas	Só rede ou só identidade; sem certs/segredos	Não cobre rede/segurança
Custo Financeiro e Operacional	Egresso, storage, compute destino, LB/rede; horas-equipe; cronograma; riscos	Estimativas presentes; premissas parciais	Custos em alto nível; sem esforço operacional	Sem estimativas
Validação e Testes de DR	Plano de testes (simulado); critérios de sucesso; evidências	Testes descritos; critérios básicos	Apenas menção a teste	Sem testes

Anexo A — Rubrica Detalhada (com Estágios, Evidências e Erros Comuns)

Como ler a rubrica

Cada critério está alinhado aos Objetivos de Aprendizagem e subdividido por nível. Para **aprovação**, o grupo deve atingir pelo menos o nível **Regular (2)** em todos os critérios. Níveis superiores exigem integração, documentação e boas práticas adicionais. Em cada estágio listamos: *escopo mínimo*, *evidências obrigatórias* e *erros comuns*.

Objetivos de Aprendizagem (referência) (1) Conceitos de nuvem e sistemas distribuídos. (2) Administração/provisionamento no console. (3) Desenvolvimento com escalabilidade. (4) Projeto, segurança e SLA/custos.

Estágio 1 — Estrutura mínima (Obj. 1 e 2)

Regular (2) — Escopo mínimo para passar

Site estático no S3 (páginas: Empresa, Grupo, Área de Clientes); EC2 acessível com `/health`; VPC criada; SGs básicos.

- **Evidências:** prints do S3 (objects/website hosting), EC2 em execução (status checks ok), `/health` retornando 200.
- **Erros comuns:** bucket site sem `index.html/404.html`; EC2 com porta errada no SG; instância sem role (dificulta etapas seguintes).

Bom (3)

VPC com sub-redes públicas/privadas; ALB já configurado (mesmo provisório); documentação parcial do desenho de rede.

- **Evidências:** diagrama simples de VPC; ALB DNS ativo (responde 4xx/5xx, healthchecks verdes ou configurados).

Excelente (4)

Infra completa e coerente: VPC (2 AZs), IGW, NAT; SGs organizados; EC2 com user-data e documentação clara das decisões.

- **Evidências:** tabela de sub-redes/rotas; justificativa de cidr; prints de rotas; `/health` publicado via ALB (mesmo que temporário).

Estágio 2 — Upload e Conversão (Lambda + DynamoDB) (Obj. 2 e 3)

Regular (2) — Escopo mínimo para passar

Upload pelo navegador usando URL pré-assinada; evento `ObjectCreated` aciona Lambda; Lambda converte para Base64 e grava no DynamoDB (único item ou poucos chunks).

- **Evidências:** prints de: (i) geração da URL pré-assinada; (ii) `ObjectCreated` no S3; (iii) execução do Lambda (CloudWatch Logs); (iv) item no DynamoDB com Base64.
- **Erros comuns:** URL expirada; `ContentType` não validado; Lambda sem permissão de `s3:GetObject` ou `dynamodb:PutItem`; strings Base64 muito grandes em um único item (estourando 400 KB).

Bom (3)

Metadados gravados (contentType, sizeBytes, timestamps); chunking implementado para imagens maiores; documentação do fluxo (passo a passo com prints).

- **Evidências:** item META e itens 000000...; cálculo de #chunks; explicação de limites do DynamoDB.

Excelente (4)

Fluxo robusto e auditável: validações de tipo/tamanho; logs significativos; tratamento de erro (ex.: ignora tipos inválidos); limpeza opcional do objeto bruto após conversão.¹

- **Evidências:** prints de logs com imageId, totalChunks, sizeBytes, e tentativa de upload inválido rejeitada pelo Lambda.

Estágio 3 — Área de Clientes + Escalabilidade (Obj. 3)

Regular (2) — Escopo mínimo para passar

Área de clientes acessível via ALB (HTTPS recomendado). Listagem básica das imagens; endpoint que retorna Base64 de uma imagem por imageId.

- **Evidências:** GET /api/images/{id} retorna JSON com base64; página da área de clientes chama e imprime no console.
- **Erros comuns:** CORS ausente; SG do EC2 sem permitir tráfego do ALB; healthcheck do Target Group apontando para rota inexistente.

Bom (3)

ALB + ASG com 2–4 instâncias; *graceful shutdown*; health checks corretos; documentação descrevendo política de escalonamento (ex.: target tracking).

- **Evidências:** gráfico do ASG (desired x inService); ALB Target Health verde em múltiplas AZs; prints de Listener 443 com ACM.

Excelente (4)

Evidência de teste de carga (leve) mostrando continuidade do serviço durante scale-out; documentação com diagramas atualizados (VPC, dados, fluxo de eventos).

- **Evidências:** série temporal de requisições/latência; registro do momento de scale-out; captura de logs de /health nas novas instâncias.

Estágio 4 — Projeto Avançado (Segurança, SLA e Custo) (Obj. 4)

Regular (2) — Escopo mínimo para passar

Relatório final descrevendo arquitetura, escolhas de serviços e funcionamento de ponta a ponta; menção a segurança básica (HTTPS, SGs restritos).

- **Evidências:** diagrama geral; prints-chave do console; explicação sucinta de por que S3/Lambda/DynamoDB.

Bom (3)

Análise de custo (metodologia via AWS Pricing/Calculator), riscos e limitações; mapeamento de permissões (IAM) de EC2 e Lambda; breve plano de contingência.

¹A limpeza é opcional e depende do que você quer avaliar.

- **Evidências:** planilha/print do Calculator; tabela de políticas e papéis; matriz risco/mitigação.

Excelente (4)

Documento completo com: diagrama lógico e físico; discussão de SLA dos serviços (citando fontes), controle de orçamento (AWS Budgets) e *hardening* adicional (bloqueio público S3, versões, logs).

- **Evidências:** criação de Budget com alerta; política pública apenas no site estático; uploads privados com URL assinada; versionamento em buckets.

Anexo B — Guia Detalhado do Plano de DR / Cloud Exit e Rubrica

B.1 Objetivo do Anexo

Este anexo explica em profundidade a rubrica específica do **Plano de Disaster Recovery (DR) e Cloud Exit**, definindo: (i) o que deve constar no documento oficial de DR; (ii) as *evidências mínimas* por critério; (iii) erros comuns a evitar; e (iv) como cada nível (Insuficiente, Regular, Bom, Excelente) será atribuído.

Regra de aprovação O grupo deve alcançar ao menos **Regular (2)** em **todos** os critérios deste plano para aprovação geral do projeto.

B.2 Estrutura exigida do Documento Oficial de DR

O documento deve ser escrito em tom formal e conter, no mínimo, as seções abaixo (pode ser um arquivo separado “Plano_DR.tex”):

1. **Escopo, Objetivo e Premissas** (contexto, decisão de saída, nuvem de destino).
2. **Objetivos de Recuperação (RTO/RPO) e Cenários de Falha** (por serviço).
3. **Arquitetura-Alvo na Nuvem de Destino** (desenho lógico e breve descrição).
4. **Mapeamento de Serviços AWS → Destino** (tabela e observações de portabilidade).
5. **Procedimentos (Runbooks)**: preparação, ativação, validação, rollback e comunicação.
6. **Rede, Segurança e Identidade** (VPC/VNet, SG/NSG, IAM/RBAC, certificados).
7. **Custos da Transição** (financeiros e operacionais, com premissas e cronograma).
8. **Validação e Testes de DR** (casos, critérios de aceite, evidências).
9. **Riscos e Compliance** (probabilidade, impacto, mitigação; LGPD).
10. **Responsáveis e Cronograma** (RACI simplificado).

B.3 Critério 1 — Objetivos e Estratégia (RTO/RPO & cenários)

Entregável obrigatório: seção com *RTO* (tempo máximo de indisponibilidade tolerado) e *RPO* (perda máxima de dados tolerada) **por serviço** (site estático, API/EC2, Lambda, DynamoDB, S3, ALB/DNS), além de *cenários de falha* (ex.: indisponibilidade regional, falha prolongada de serviço, decisão estratégica).

Evidências mínimas:

- Tabela de RTO/RPO por serviço e justificativas.
- Lista de cenários de falha e estratégia escolhida (cold/warm/active-passive) com trade-offs.

Erros comuns a evitar: RTO/RPO genéricos “para tudo”; ausência de justificativa; ignorar DNS/ACM.

Avaliação:

- *Insuficiente (1)*: sem RTO/RPO ou cenários.
- *Regular (2)*: RTO/RPO em alto nível e ao menos 1 cenário descrito.
- *Bom (3)*: RTO/RPO definidos por serviço e principais cenários; justificativa parcial.
- *Excelente (4)*: RTO/RPO por serviço com trade-offs claros e escolha de estratégia (cold/warm/active) coerente com o caso.

B.4 Critério 2 — Mapeamento de Serviços & Portabilidade

Entregável obrigatório: tabela mapeando os componentes AWS para equivalentes na nuvem alvo e observações de portabilidade/limites.

AWS	Azure (equiv.)	GCP (equiv.)	Observações de portabilidade
S3 (uploads/site)	Blob Storage (+ SAS)	Cloud Storage (+ Signed URL)	Assinaturas (SAS/URL assinada); políticas de acesso
Lambda	Azure Functions	Cloud Functions	Trigger de Storage; runtime; variáveis de ambiente
DynamoDB	Cosmos DB (API NoSQL)	Firestore/Datastore	Chave-partição; limites; migração de chunks Base64

Evidências mínimas: tabela preenchida.

Avaliação:

- *Insuficiente (1)*: sem mapeamento.
- *Regular (2)*: mapeamento mínimo (storage, compute, função) sem detalhes.
- *Bom (3)*: maioria dos serviços mapeados; riscos principais descritos.
- *Excelente (4)*: **todos** mapeados (incl. IAM/DNS/certs) com riscos/limites e mitigação.

B.5 Critério 3 — Procedimentos e Runbooks

Entregável obrigatório: passo a passo numerado para **Preparação, Ativação (em crise), Validação, Rollback e Comunicação**. Indicar pré-requisitos, responsáveis e critérios de sucesso.

Evidências mínimas:

- Lista ordenada de passos com pré-reqs.
- Checklist de validação e plano de rollback.
- Plano de comunicação (quem avisa, quando, como).

Avaliação:

- *Insuficiente (1)*: incompleto/confuso.
- *Regular (2)*: alto nível, sem validação/rollback estruturados.
- *Bom (3)*: passos claros + validações básicas.
- *Excelente (4)*: passos claros + validações e rollback e comunicação definidos. 6 enditemize

B.6 Critério 5 — Rede, Segurança e Identidade

Entregável obrigatório: mapeamento VPC, IAM, roteamento; política de menor privilégio e segregação de ambientes.

Evidências mínimas:

- Desenho lógico de rede no destino (sub-redes, balanceadores).
- Planos de certificados (emissão/rotação) e *secrets management*.

Avaliação:

- *Insuficiente (1)*: não cobre rede/segurança.
- *Regular (2)*: cobre superficialmente rede *ou* identidade.
- *Bom (3)*: mapeia itens principais; poucas lacunas.
- *Excelente (4)*: cobre rede, identidade e TLS/segredos de forma completa.

B.7 Critério 6 — Custo Financeiro e Operacional

Entregável obrigatório: estimativa com **premissas** explícitas (egresso, armazenamento destino, compute/load balancer destino) e **esforço operacional** (horas-equipe por papel), **cronograma** e **riscos**.

Evidências mínimas:

- Tabela com custos diretos (egresso, storage, compute, rede/LB) e indiretos (horas-equipe, treino, ferramentas).
- Premissas quantitativas (ex.: GB a migrar, req/s, horas de janela).

Erros comuns: custos só qualitativos; ignorar egresso; sem esforço de equipe.

Avaliação:

- *Insuficiente (1)*: sem estimativas.
- *Regular (2)*: custos em alto nível; sem esforço operacional.
- *Bom (3)*: estimativas com premissas; esforço parcial.
- *Excelente (4)*: estimativas + esforço e cronograma e riscos.

B.8 Critério 7 — Validação e Testes de DR

Entregável obrigatório: plano de testes de DR (simulado), **critérios de sucesso** e **evidências**.

Evidências mínimas:

- Casos: upload → conversão → recuperação Base64 no destino.
- Critérios: resposta 200, integridade amostrada, latência aceitável, DNS cutover validado.
- Prints/relatórios: logs da função, métricas do balanceador, verificação de imagem reconstituída (data: URL).

Avaliação:

- *Insuficiente (1)*: sem plano de testes.
- *Regular (2)*: menção a teste, sem critérios claros.
- *Bom (3)*: testes descritos com critérios básicos.
- *Excelente (4)*: testes com critérios e evidências e checklist de aceite.

Observação final O nível **Regular (2)** requer presença e coerência mínima em todos os itens acima. Níveis maiores demandam profundidade, completude e evidências robustas.