ANONIMIZAÇÃO E PRÉ-PROCESSAMENTO DE EXAMES DE ELETROCARDIOGRAMAS

Douglas Aldred [1]; Paulo Pirozelli [2]

[1] Aluno de Iniciação Científica do Instituto Mauá de Tecnologia (IMT);
[2] Professor do Instituto Mauá de Tecnologia (IMT).

Resumo

O eletrocardiograma (ECG) é um dos exames mais utilizados na prática clínica para detecção de anomalias cardiovasculares. Este trabalho descreve o desenvolvimento de um pipeline para anonimização e pré-processamento de uma base com 1,5 milhão de ECGs obtidos da Beneficência Portuguesa de São Paulo. O pipeline foi projetado para eliminar informações sensíveis de forma automatizada, garantindo privacidade, qualidade e escalabilidade. A solução, implementada em Python e executada em ambiente AWS ECS Fargate (ARM64), combina processamento assíncrono e paralelo para lidar com operações I/O-bound e CPU-bound de maneira eficiente. O sistema processou aproximadamente 1,5 milhão de exames com zero falhas, atingindo desempenho médio de 5 arquivos por segundo, tempo total de 83 horas e custo de US$69 (US$0,000046 por exame). O pipeline realiza a anonimização de campos identificáveis e a rasterização das páginas dos exames em alta resolução (450–600 DPI), aumentando a qualidade diagnóstica das imagens para posterior uso em modelos de deep learning. Em comparação com uma abordagem baseada em AWS Lambda e Step Functions, a arquitetura proposta reduziu o custo em cerca de 22% e o tempo de execução em 17%, além de garantir maior controle sobre concorrência e reutilização de recursos. A metodologia proposta oferece uma solução reprodutível, eficiente e economicamente sustentável para o tratamento de dados médicos em larga escala, contribuindo para o avanço de pesquisas em inteligência artificial aplicada à saúde.

Introdução

Doenças cardiovasculares representam a principal causa de mortalidade no mundo, sendo responsáveis por cerca de um terço dos óbitos globais (World Heart Federation, 2023). No Brasil, essas enfermidades figuram entre as principais causas de internação hospitalar, evidenciando a necessidade de diagnósticos rápidos e precisos (Linz *et al*., 2024). Nesse contexto, o eletrocardiograma (ECG) representa uma ferramenta essencial para a detecção precoce de anomalias cardíacas. O ECG é um exame que registra a atividade elétrica do coração ao longo do tempo por meio de eletrodos posicionados na pele — sendo não invasivo, acessível e de baixo custo (Lopes *et al*., 2019). Contudo, o aumento do volume de exames e a escassez de especialistas tornam a análise manual lenta e suscetível a erros. Nesse cenário, a inteligência artificial (IA) e o aprendizado profundo (*deep learning*) surgem como soluções promissoras para automatizar a interpretação de ECGs e apoiar decisões clínicas (Ribeiro *et al*., 2020; Chang *et al*., 2021). Diante desse panorama, pesquisas nacionais que unem engenharia de dados e IA tornam-se fundamentais para viabilizar diagnósticos automatizados e escaláveis dentro do Sistema Único de Saúde (SUS).

Em parceria entre o Instituto Mauá de Tecnologia (IMT) e a Beneficência Portuguesa de São Paulo (BP), foi desenvolvida uma base de dados com mais de 1,5 milhão de exames de ECG — uma das maiores já reunidas no Brasil — destinada ao desenvolvimento de modelos de deep learning aplicados à saúde. Entretanto, para viabilizar seu uso, é necessário um pipeline rigoroso de engenharia de dados que assegure a anonimização de informações sensíveis nas imagens e metadados (ex.: nome do paciente, CRM do médico e assinaturas manuscritas), além de etapas de pré-processamento de imagem. O pipeline proposto aplica tarjas automáticas em

áreas sensíveis e técnicas de melhoria da qualidade da imagem, como realce, equalização e redução de ruído, preservando a integridade visual necessária para a extração de *features* (Bera *et al*., 2022; Basu *et al*., 2023; Dias *et al*., 2023). O objetivo deste trabalho é apresentar a metodologia de anonimização e pré-processamento desenvolvida para viabilizar o uso seguro e eficiente dessa base de ECGs em aplicações de IA médica.

Mais do que uma aplicação direta de IA, este estudo evidencia o papel da engenharia de dados na construção de pipelines escaláveis, de baixo custo e adequadas ao contexto clínico. O processamento foi otimizado em uma infraestrutura em nuvem (AWS), com uso de paralelismo, processamento assíncrono e *multithreading* para lidar com vários arquivos de forma eficiente. Estratégias de compressão e padronização entre 450 e 600 DPIs foram adotadas para preservar detalhes clínicos, reduzir custos de armazenamento e manter a qualidade necessária ao treinamento de modelos avançados (Dong *et al*., 2023; Dias *et al*., 2023).

Material e Métodos

O sistema foi desenvolvido em Python 3.12 e executado em uma única instância AWS ECS Fargate, sobre arquitetura ARM64 (família Graviton). Essa instância concentra todo o processamento, explorando paralelismo e concorrência dentro da própria máquina. O objetivo é garantir alto desempenho sem a necessidade de múltiplas instâncias, reduzindo custo e complexidade operacional. O pipeline é responsável por processar lotes de exames armazenados no Amazon S3, aplicando anonimização e otimização de imagem antes do reenvio dos resultados. A arquitetura segue um modelo híbrido de concorrência assíncrona e paralelismo real, permitindo lidar eficientemente com operações de entrada e saída (I/O-bound) e tarefas computacionalmente intensivas (CPU-bound) em uma única instância.

As operações do sistema são estruturadas em quatro etapas: (1) listagem e enfileiramento de arquivos, (2) leitura e pré-processamento, (3) anonimização e (4) compressão (ZIP) e reenvio. Operações de I/O, como leitura e escrita em S3, utilizam o modelo assíncrono do módulo *asyncio*, que opera sobre o *event loop* do sistema operacional apoiado em mecanismos como epoll no Linux para monitorar diversos fluxos simultaneamente sem bloqueio de execução. Essa abordagem reduz o *context switching* e permite a transferência de centenas de arquivos em paralelo com baixo consumo de recursos (Tanenbaum e Bos, 2015; Silberschatz *et al*., 2020). Já as tarefas CPU-bound, como rasterização, anonimização e compressão, utilizam ProcessPoolExecutor, que distribui o trabalho entre os núcleos físicos da instância, garantindo paralelismo real e aproveitamento integral da capacidade de processamento. O sistema mantém um equilíbrio dinâmico entre CPU e memória, evitando sobrecarga e garantindo máxima eficiência no throughput de arquivos processados.

O núcleo da anonimização, realiza a remoção de informações sensíveis combinando análise textual e redaction espacial. A primeira página do exame é processada vetorialmente com PyMuPDF, detectando rótulos como “Nome”, “CRM” e “Responsável”, e aplicando tarjas sobre seus respectivos valores. A segunda página, que contém o traçado eletrocardiográfico, é rasterizada em alta resolução (450–600 DPI) e convertida em imagem via Pillow (PIL), na qual são aplicadas redactions retangulares em regiões predefinidas. Essa abordagem garante anonimização total sem degradação perceptível do sinal, requisito fundamental para posterior uso dos dados em modelos de aprendizado profundo. Todas as operações intensivas são executadas em paralelo, explorando o máximo de desempenho de cada núcleo da CPU.

Para a implantação, o sistema foi containerizado com Docker, garantindo portabilidade e consistência entre ambientes. O contêiner foi configurado para execução em arquitetura ARM64, mesmo sendo desenvolvido em ambiente x86-64. Essa conversão foi realizada

utilizando o Docker Buildx, ferramenta que permite *cross-compilation* e emulação via QEMU, gerando imagens ARM compatíveis com instâncias Graviton da AWS (Docker Inc., 2024). Esse processo assegura que a mesma imagem possa ser construída e testada localmente em máquinas x86 e posteriormente executada em ambiente ARM sem necessidade de recompilação. O uso de contêineres também simplifica o controle de dependências e garante reprodutibilidade total do ambiente de execução.

A escolha pela arquitetura ARM64 (Graviton) deve-se à sua maior eficiência energética e melhor relação custo-benefício quando comparada a arquiteturas x86, especialmente em cargas de processamento intensivo. Enquanto instâncias x86 apresentam maior consumo energético e custo por vCPU, os processadores Graviton oferecem desempenho equivalente ou superior com menor consumo, resultando em redução significativa de custos operacionais e melhor aproveitamento por watt. Segundo a AWS, as instâncias Graviton oferecem até 40% de melhor preço-performance em comparação às instâncias baseadas em x86 (AWS, 2024). Essa vantagem decorre de um design otimizado para múltiplos núcleos e do uso eficiente de instruções vetoriais SIMD NEON, amplamente utilizadas por bibliotecas como libjpeg-turbo. Essa biblioteca, empregada no Pillow para compressão de imagens, utiliza vetorização NEON em processadores ARM, obtendo aceleração de 2 a 6 vezes em relação à implementação padrão do libjpeg (libjpeg-turbo, 2024). Essa sinergia entre hardware e software proporciona significativa redução no tempo de rasterização e compressão, mantendo alta qualidade visual e reduzindo o custo total de processamento

Ao término do processamento, os arquivos anonimizados são recombinados, comprimidos no formato ZIP Deflate (nível 5) e reenviados ao bucket de saída no S3, acompanhados de metadados como tamanho original, taxa de compressão e identificador único (ULID). A metodologia empregada alia fundamentos de engenharia de dados, sistemas operacionais, computação paralela e virtualização de contêineres, resultando em uma esteira altamente otimizada, reprodutível e eficiente, capaz de processar milhões de exames com anonimização total e preservação da qualidade diagnóstica (Dias *et al*., 2023; Dong *et al*., 2023).

Resultados e Discussão

A execução do pipeline de anonimização foi realizada em uma instância AWS ECS Fargate com arquitetura ARM64 (Graviton), configurada com 8 vCPUs e 16 GB de memória. O sistema processou aproximadamente 1,5 milhão de exames de ECGs armazenados no Amazon S3, atingindo uma taxa média de 5 arquivos por segundo, ou cerca de 18.000 exames por hora. O tempo total de execução do lote completo foi de aproximadamente 83 horas, com desempenho estável e sem ocorrência de falhas durante todo o processamento. A Figura 1 apresenta o fluxo completo da aplicação.

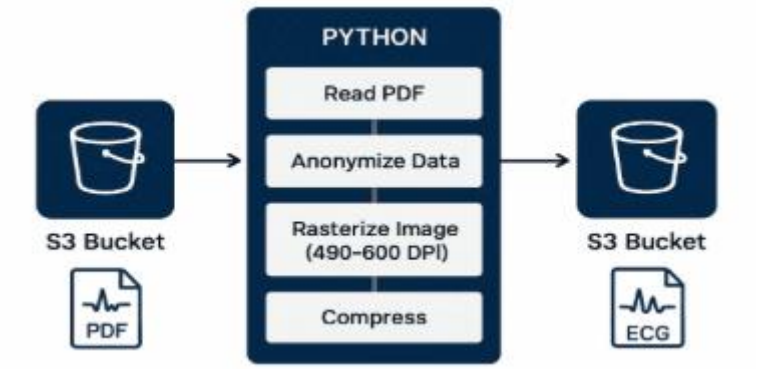

Figura 1 - Fluxo completo entre o Amazon S3 e o módulo Python responsável por ler o PDF, anonimizar os dados sensíveis, rasterizar a imagem em alta resolução (490–600 DPI) e comprimir o arquivo processado antes de enviá-lo ao bucket de saída.

A arquitetura geral do pipeline é estruturada em três blocos principais: o bucket de entrada no Amazon S3, o módulo de processamento em Python executado no ECS Fargate e o bucket de saída. Cada arquivo PDF é extraido do armazenamento, processado de forma independente e retornado de maneira anonimizada, garantindo um fluxo contínuo entre entrada e saída. A arquitetura geral do pipeline é estruturada em três blocos principais: o bucket de entrada no Amazon S3, o módulo de processamento em Python executado no ECS Fargate e o bucket de saída.

No módulo Python, as etapas executadas incluem leitura do PDF, remoção de informações sensíveis, rasterização da página com o traçado do ECG em alta resolução e compressão final. Essa organização modular permite que cada etapa seja paralelizada e controlada de forma eficiente, o que é refletido nos resultados apresentados na sequência da seção.No módulo Python, as etapas executadas incluem leitura do PDF, remoção de informações sensíveis, rasterização da página com o traçado do ECG em alta resolução e compressão final. Essa organização modular permite que cada etapa seja paralelizada e controlada de forma eficiente, o que é refletido nos resultados apresentados na sequência da seção.

O pipeline foi configurado para operação de alta concorrência controlada, combinando 200 workers assíncronos para operações de entrada e saída (*I/O-bound*) com 16 workers paralelos de CPU (*CPU-bound*). A fila de tarefas (QUEUE_SIZE = 800) garantiu equilíbrio entre throughput e uso de memória, evitando bloqueios e saturação da instância. Essa arquitetura permitiu que o sistema mantivesse uso contínuo de CPU acima de 90%, aproveitando integralmente os recursos da instância ARM64.

O custo operacional total estimado foi de US$69 para o processamento completo, considerando 83 horas de execução e custo médio de US$0,83/hora da instância Fargate. Isso representa um custo unitário de US$0,000046 por exame, evidenciando a eficiência econômica e computacional do pipeline. A Tabela 1 apresenta as principais métricas obtidas, demonstrando a consistência da execução e a viabilidade prática da abordagem proposta para anonimização em larga escala.

Métrica	Valor médio	Observação
Taxa de processamento (PDF/s)	5	Média estável com 200 workers assíncronos
Arquivos processados por hora	18 000	Escalabilidade linear até 8 vCPUs
Processos paralelos de CPU	16	Configuração MAX_PROCESS_WORKERS
Workers assíncronos de I/O	200	Configuração MAX_WORKERS
Tamanho máximo da fila	800	Configuração QUEUE_SIZE
Tempo total para 1,5 M arquivos	~83 h	Execução contínua sem interrupções
Tamanho de entrada (médio)	300 KB	PDFs originais com baixa resolução
Tamanho de saída (médio)	3 MB	PDFs rasterizados em alta qualidade (450 DPI)
Custo total estimado (USD)	69	Baseado em 83 horas de execução
Custo médio por exame (USD)	0,000046	Custo unitário extremamente baixo
Erros de processamento (%)	0	Nenhum arquivo inválido detectado

Tabela 1 – Métricas operacionais do pipeline, incluindo throughput, parâmetros de concorrência (I/O e CPU), características dos arquivos processados e custos agregados observados durante a execução completa dos 1,5 milhão de exames.

Durante o processamento, observou-se que o tamanho médio dos arquivos aumentou de ~300 KB (entrada) para ~3 MB (saída) após o processo de rasterização e melhoria da qualidade da imagem. Esse aumento está relacionado à conversão da segunda página dos exames para imagens em alta resolução (450–600 DPI) e compressão JPEG com qualidade de 95%, necessária para preservar a integridade dos traçados eletrocardiográficos. Embora isso represente um aumento de aproximadamente 10x no tamanho final, tal escolha é justificada pelo ganho em fidelidade visual, essencial para análises que fazem uso de modelos de aprendizado profundo.

Antes da consolidação da arquitetura atual, foi desenvolvida uma versão baseada em AWS Lambda e Step Functions, projetada para processar arquivos em paralelo com até 500 invocações simultâneas. Cada função Lambda processava em média 6 a 8 PDFs por execução, com tempo total de 14 minutos, próximo ao limite máximo permitido (900s). Essa abordagem utilizava 10GB de memória por execução e resolução de 300 DPI, com cada função sendo responsável por baixar, anonimizar e reenviar o arquivo ao S3.

Apesar da escalabilidade imediata, essa arquitetura apresentou limitações práticas, principalmente em custo e eficiência de CPU. Como cada Lambda reinicializava o ambiente e carregava dependências a cada invocação, o overhead de inicialização impactava a performance global. Além disso, o custo médio estimado foi de aproximadamente US$850, superior ao custo

médio de US$660 obtido na versão Fargate, mesmo com maior qualidade e controle de processamento.

A tabela abaixo apresenta o comparativo consolidado entre as duas implementações, destacando as diferenças de desempenho, custo e qualidade.

Parâmetro	AWS Lambda + Step Functions	AWS ECS Fargate (ARM64)
Arquitetura de execução	Stateless, invocações efêmeras	Contêiner persistente e controlado
Concorrência máxima	500 funções simultâneas	200 workers assíncronos + 16 processos paralelos
PDFs processados por execução	6–8 por Lambda	1,5 milhão no lote contínuo
Tempo total para 1,5 M arquivos	~100 h	~83 h
Tempo por execução (lote)	~14 min	Execução contínua
DPI de renderização	300	450–600
Uso de memória	~10 GB por função	16 GB compartilhados entre processos
Custo mensal estimado (USD)	850	660
Custo por 1,5 M arquivos (USD)	~89	~69
Custo médio por exame (USD)	0,000059	0,000046
Erros de processamento (%)	0,1 (variação)	0
Qualidade da imagem	Média	Alta (maior fidelidade)
Eficiência energética (ARM64)	Não aplicável (x86)	Alta (Graviton, +40% eficiência)

Tabela 2 – Comparação técnica entre as arquiteturas AWS Lambda + Step Functions e AWS ECS Fargate (ARM64), incluindo limites de concorrência, eficiência de execução, uso de recursos, qualidade da imagem e custo operacional no processamento dos 1,5 milhão de exames.

A análise comparativa mostra que a arquitetura Lambda + Step Functions, embora vantajosa pela simplicidade e paralelização automática, apresentou custo mais elevado e menor eficiência para cargas *CPU-bound*. A arquitetura ECS Fargate, por outro lado, consolidou o processamento em uma única instância ARM64, permitindo controle direto da concorrência, reuso de processos e melhor aproveitamento da memória. O uso contínuo dos recursos, aliado à ausência de reinicializações a cada tarefa, resultou em uma redução de custo de aproximadamente 22% e aumento de qualidade de imagem, sem comprometer o tempo total de processamento.

A figura 2 apresenta o resultado final da anonimização aplicada ao conjunto de 1,5 milhão de exames. Na parte de cima, é possível observar a remoção completa dos campos identificáveis por redactions vetoriais, enquanto na parte o exame é mostrado após o processo de rasterização em alta resolução, mantendo a fidelidade diagnóstica do traçado eletrocardiográfico. Os exemplos ilustram que o algoritmo executa de forma consistente todas as etapas de anonimização, remoção de textos sensíveis, redactions posicionais e limpeza de metadados, preservando a parte essencial do exame, que é o traçado eletrocardiográfico e outras informações como a amplitude das ondas e o laudo contido no exame

Figura 2 – Exemplo de saída do pipeline de anonimização. Primeira página do exame após remoção completa dos dados sensíveis, incluindo nome, CPF, data de nascimento, RG, informações administrativas e identificações de profissionais. Segunda página após rasterização em alta resolução (450–600 DPI) e aplicação de redactions por coordenadas, preservando integralmente o traçado eletrocardiográfico.

Esses resultados demonstram que, do ponto de vista científico e operacional, a abordagem ECS Fargate é mais adequada para cargas contínuas e datasets médicos em larga escala. Ela oferece maior eficiência energética, controle previsível de custos, processamento estável e mantém os princípios de reprodutibilidade e padronização, essenciais em pipelines de engenharia de dados aplicados à saúde. Além da validação automática realizada para cada arquivo, verificando anonimização, rasterização e reconstrução do PDF, conduzimos também uma verificação manual por amostragem. Os exames inspecionados aleatoriamente não apresentaram falhas de anonimização ou perda de conteúdo clínico, confirmando a confiabilidade do processo como um todo.

Conclusões

A solução desenvolvida demonstrou que é possível anonimizar em larga escala exames de ECG de forma automatizada, eficiente e economicamente viável, preservando a integridade visual necessária para análises médicas e aprendizado profundo. Por meio de uma arquitetura baseada em AWS ECS Fargate (ARM64), o pipeline atingiu desempenho de 5 arquivos por segundo, processando 1,5 milhão de exames em aproximadamente 83 horas, sem falhas e com custo total de apenas US$69.

Essa implementação consolidou uma abordagem híbrida de concorrência assíncrona e paralelismo real, capaz de equilibrar operações *I/O-bound* e *CPU-bound* em uma única instância, explorando integralmente os recursos computacionais disponíveis. O uso de processos paralelos, fila controlada de tarefas e compressão inteligente viabilizou a execução contínua e estável do sistema, reduzindo sobrecarga e garantindo previsibilidade operacional. A rasterização em alta resolução (450–600 DPI) elevou o tamanho médio dos arquivos de 300 KB para 3 MB, mas proporcionou imagens com fidelidade diagnóstica significativamente superior, requisito essencial para a etapa posterior de extração de features e treinamento de modelos de aprendizado profundo.

Em contraposição, a arquitetura alternativa, baseada em AWS Lambda e Step Functions, mostrou-se funcional, porém limitada. Apesar de alcançar paralelização imediata com até 500 execuções simultâneas, cada Lambda processava de 6 a 8 arquivos em média, resultando em ~100 horas totais de execução e custo estimado de US$850. O caráter efêmero das funções e o reinício do ambiente a cada invocação geraram sobrecarga de inicialização e desperdício de recursos, restringindo a eficiência em tarefas intensivas de CPU.

Portanto, o método proposto resolve de forma direta o problema inicial, a anonimização massiva e eficiente de exames médicos contendo dados sensíveis, garantindo privacidade, escalabilidade e integridade dos traçados. Além de superar as limitações de custo e desempenho das soluções anteriores, a arquitetura desenvolvida oferece um caminho reprodutível e sustentável para o processamento de grandes bases clínicas. Essa integração entre engenharia de dados, sistemas distribuídos e ética computacional representa uma contribuição concreta para o avanço da ciência de dados aplicada à saúde, fornecendo uma base sólida para o treinamento de modelos de inteligência artificial e diagnóstico automatizado de ECGs.

Referências Bibliográficas

AWS (2024). *Graviton Processors – Price Performance and Energy Efficiency Benchmarks.* Amazon Web Services. Disponível em: https://aws.amazon.com/ec2/graviton.

Basu, S. et al. (2023). *Medical image analysis using deep learning algorithms.* Frontiers in Artificial Intelligence.

Bera, D. et al. (2022). *Image-based deep learning in 12-lead ECG diagnosis.* Frontiers in Artificial Intelligence.

Chang, K.-C. et al. (2021). *Usefulness of machine learning-based detection and classification of cardiac arrhythmias with 12-lead electrocardiograms.* Canadian Journal of Cardiology.

Dias, F. M. et al. (2023). *AI-driven screening system for rapid image-based classification of 12-lead ECG exams: a promising solution for emergency room prioritization.* IEEE Access.

Docker Inc. (2024). *Docker Buildx and QEMU Emulation for Cross-Architecture Builds.* Docker Documentation. Disponível em: https://docs.docker.com/build/buildx

Dong, Y. et al. (2023). *An arrhythmia classification model based on Vision Transformer with deformable attention.* Micromachines.

libjpeg-turbo Project (2024). *Performance and SIMD Acceleration (MMX, SSE, AVX2, NEON).* Disponível em: https://www.libjpeg-turbo.org/About/Performance.

Linz, D. et al. (2024). *Atrial fibrillation: epidemiology, screening and digital health.* The Lancet Regional Health – Europe.

Lopes, M. A. C. Q. et al. (2019). *Guideline of the Brazilian Society of Cardiology on telemedicine in cardiology.* Arquivos Brasileiros de Cardiologia.

Ribeiro, A. L. P. et al. (2020). *Automatic diagnosis of the 12-lead ECG using a deep neural network.* Nature Communications.

Silberschatz, A.; Galvin, P.; Gagne, G. (2020). *Operating System Concepts.* 10ª ed. Wiley.

Tanenbaum, A. S.; Bos, H. (2015). *Modern Operating Systems.* 4ª ed. Pearson.

World Heart Federation (WHF). (2023). *World Heart Report 2023: Confronting the World's Number One Killer*. Geneva.