Implementação de Arquitetura Modular para Integração de Classificador no Aplicativo CitoFocus

Danilo César S. Soares[1], Gabriel M. S. Ferreira[1], Alan Erse[2] Breno N. S. Keller[1], Mariana T. Rezende[3], Claudia M. Carneiro[3], Andrea G. Campos[1]

[1]Departamento de Computação (DECOM) – Universidade Federal de Ouro Preto (UFOP)
Ouro Preto – MG – Brazil

[2]Questrade Financial Group
Belo Horizonte, MG, Brasil

[3]Escola de Fármacia (EFAR) – Universidade Federal de Ouro Preto (UFOP)
Ouro Preto – MG – Brazil

danilo.soares@aluno.ufop.edu.br

Abstract. *The article presents the integration of a classifier based on the YOLO algorithm into the CitoFocus application, which assists cytopathologists in analyzing cytological exams. To ensure scalability and simplified maintenance, the back-end was redesigned using NestJS, a hexagonal architecture, and MVC. The new structure enabled the incorporation of the classifier with minimal impact on the functionality of CitoFocus, enhancing support for professionals and contributing to the reduction of diagnostic errors in Pap smear exams.*

Resumo. *O artigo apresenta a integração de um classificador baseado no algoritmo YOLO ao aplicativo CitoFocus, que auxilia citopatologistas na análise de exames citológicos. Para garantir escalabilidade e manutenção simplificada, o back-end foi reformulado com NestJS, arquitetura hexagonal e MVC. A nova estrutura permitiu incorporar o classificador com impacto mínimo no funcionamento do CF, aumentando o apoio gerado aos profissionais, contribuindo para a redução de erros diagnósticos no exame de Papanicolaou.*

1. Introdução

O câncer cervical (CC) é o quarto tipo mais comum entre as mulheres no mundo, sendo especialmente comum em países em desenvolvimento, onde apresenta altas taxas de novos casos e mortalidade [World Health Organization 2024]. No Brasil, é o terceiro tumor maligno mais frequente e a quarta principal causa de morte por câncer em mulheres [Instituto Nacional do Câncer 2022].

Os estágios iniciais da doença raramente manifesta sintomas, tornando essencial o rastreamento precoce, pois a detecção nessa fase permite tratamentos eficazes resultando na cura. O método mais utilizado é o exame de Papanicolaou (PAP), que consiste na coleta de células do colo do útero para análise laboratorial. No entanto, a análise

dessas amostras é um processo repetitivo, exaustivo, e interpretativo, dependendo diretamente da experiência do profissional citopatologista. Devido a isso, está sujeito a erros que podem resultar em diagnósticos incorretos, sejam falsos-negativos, que atrasam o tratamento, ou falsos-positivos, que geram impactos desnecessários na vida das pacientes [Rezende et al. 2021].

Para auxiliar no processo de análise, é comum os citopatologistas buscarem trocas de conhecimento e discussão de casos, recorrendo para isso às mídias sociais. Nesse contexto, foi desenvolvido o CitoFocus (CF) [Guimarães 2021, Keller et al. 2021], um aplicativo móvel que permite o compartilhamento de casos em um ambiente especializado. Nele, profissionais podem interagir por meio de votos e justificativas, contribuindo para a análise e discussão de casos clínicos publicados.

O CF foi criado para conectar citopatologistas, facilitando a colaboração e análise de casos em um ambiente profissional. Os usuários podem criar casos, adicionar imagens e informações que auxiliam no diagnóstico sem expor os pacientes, além de participar das votações e fornecer justificativas. Para enriquecer a análise e contribuir na construção de um diagnóstico, propomos a integração de um classificador - propomos a integração de um classificador.

No entanto, essa adição representa um desafio arquitetural. Nossa abordagem prevê uma estrutura modular que permita essa integração, garantindo que o classificador contribua com mais informações para apoiar o usuário.

Assim, este trabalho tem como objetivo propor e implementar uma arquitetura modular que possibilite a integração do classificador proposto por [Diniz et al. 2022] no CF sem impactar seu funcionamento. Especificamente, busca-se desenvolver e modularizar o *back-end*, adaptar o classificador para consumo pelo aplicativo e integrá-lo ao sistema.

2. Revisão Bibliográfica

O exame citopatológico é essencial para a detecção precoce do CC, identificando lesões precursoras antes da progressão da doença [Fundação Oswaldo Cruz 2023]. O método mais utilizado é o PAP, que, apesar de eficiente, está sujeito a erros diagnósticos [World Health Organization 2024].

Nesse cenário, temos o CF, ferramenta que visa apoiar os profissionais. Com o intuito de aumentar o suporte aos profissionais, propomos a integração a um classificador. Para permitir isso, foi preciso que a arquitetura do *back-end* fosse adaptada.

Nessa reestruturação, utilizamos conceitos de modularização - desacoplamento da aplicação em pequenos contextos [Gomes et al. 2023] -, arquitetura hexagonal, cuja ideia central é isolar o núcleo da aplicação (regra de negócios) das interações externas (banco de dados, APIs, interfaces, entre outros) e MVC (Model-View-Controller), que separa o código em responsabilidades que favorecem o desenvolvimento e manutenção. Todos estes conceitos garantem escalabilidade e facilidade na revisão da aplicação.

Para a construção do *back-end*, foi utilizado o *NestJS*, *framework* baseado em *NodeJs*[1], que recomenda a construção baseada em entidades para garantir alto desacopla-

[1]https://nodejs.org/en/ Acessado em: 03/03/2025

mento e escalabilidade - potencial da aplicação receber atualizações de forma facilitada. Quanto ao classificador, a implementação proposta utiliza o *YOLO (You Only Look Once)*, algoritmo de *Machine Learning* e *Deep Learning* treinado para detecção e classificação de imagens.

Machine Learning é um ramo da Inteligência Artificial que desenvolve sistemas capazes de identificar padrões e tomar decisões com mínima interferência humana. Dentro da ML, temos a *Deep Learning*, que utiliza redes neurais artificiais compostas por várias camadas (inspiradas no funcionamento do cérebro humano) para processar dados e extrair informações de alto nível. Um exemplo disso é o próprio método YOLO, que emprega redes neurais convolucionais para detectar objetos em imagens de forma rápida e precisa.

Soluções automatizadas para análise citopatológica têm demonstrado eficiência, reduzindo falsos-negativos e falsos-positivos [Rehman et al. 2020]. Métodos como o CEA [Diniz et al. 2022] utilizam redes neurais para apoiar especialistas na classificação de células cervicais, melhorando a padronização dos diagnósticos.

Além disso, estudos sobre a integração de algoritmos de detecção em dispositivos móveis indicam a viabilidade de tais sistemas em diferentes contextos, inclusive em ambientes com recursos limitados [Iftikhar et al. 2024]. Essas abordagens destacam o potencial do uso da classificação automática para otimizar a triagem, gerando informações que apoiam os citopatologistas na construção do diagnóstico.

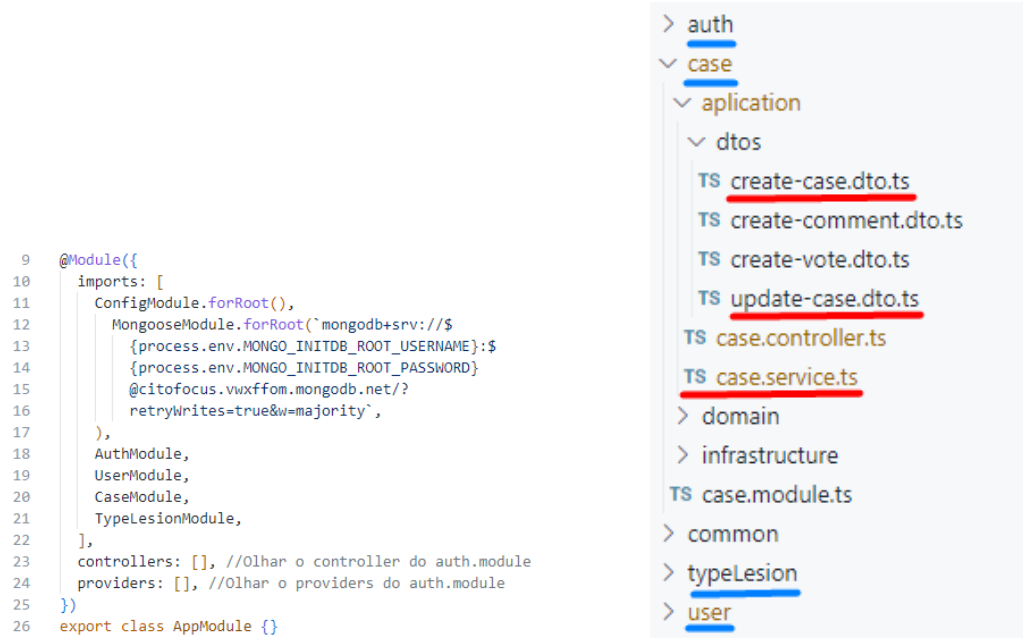
3. Método Proposto

O CF foi adaptado para permitir a colaboração remota entre citopatologistas, exigindo modificações no *back-end* para atender aos novos requisitos. Para isso, foi utilizado o *framework NestJS*, a arquitetura hexagonal e a MVC, que proporcionam modularização e flexibilidade ao sistema. Essa abordagem facilita a manutenção e evolução do sistema.

A arquitetura hexagonal define interfaces para a comunicação entre o *back-end* e o *front-end*, utilizando portas (regras de interação) e adaptadores (implementações). Os principais componentes de cada entidade incluem:

- *Domain*: define as regras de negócio das entidades, garantindo que elas sejam independentes das tecnologias externas.
- *Service*: estabelece como as rotas interagem com o *front-end*, lidando com a lógica do aplicativo.
- *Controller*: gerencia as rotas utilizadas pelo *front-end* para envio e busca de informações.
- *Repository*: implementa a comunicação com o banco de dados, permitindo a troca desse serviço com poucas alterações no código.

O sistema possui um *module* próprio, encapsulando suas regras e garantindo a modularização. Isso facilita a manutenção, pois mudanças em uma entidade não afetam as demais. Além disso, caso uma entidade precise ser removida, basta desvinculá-la do *module*, figura 1a do conjunto de figuras 1, do aplicativo sem impacto no restante do código. Com essa organização, o *back-end* do CF se tornou mais estruturado e flexível, possibilitando futuras expansões e integrações sem comprometer sua estabilidade.



(a) Arquivo module do *back-end*. **(b) Figura exibindo as entidades.**

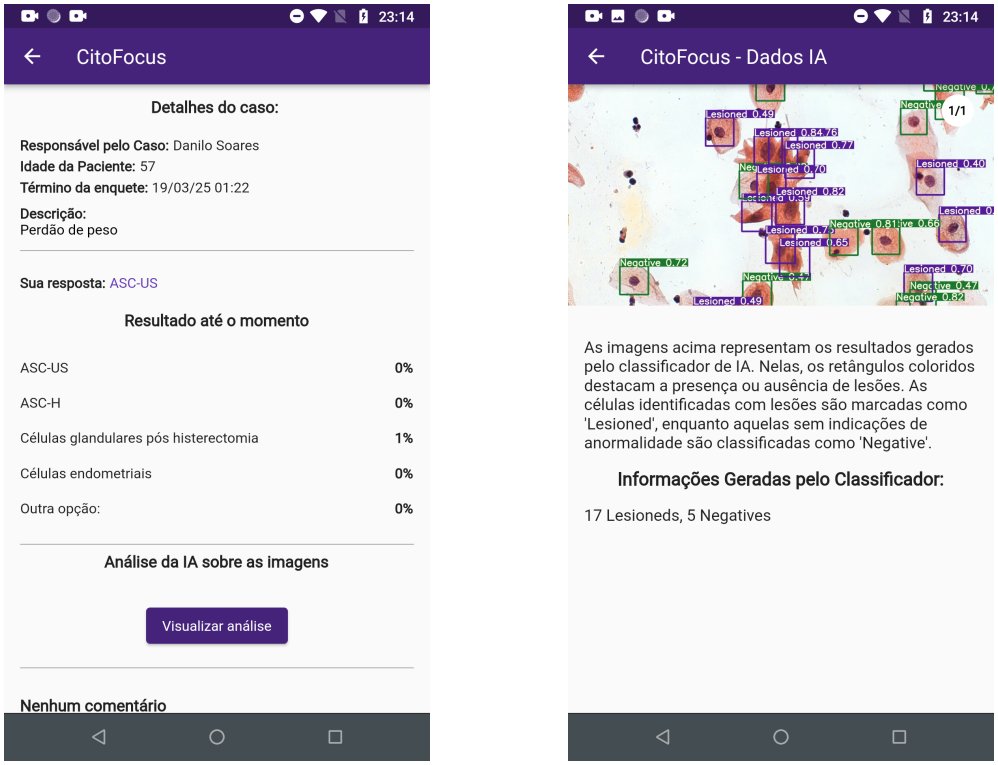
Figura 1. Figura exibindo a organização do código e dos arquivos.

Na figura 1b do conjunto de figuras 1 temos marcado em azul todas as entidades. Para a integração do classificador, foi necessário alteração somente na entidade Casos, entidade responsável pelas funções de gerenciamento de casos. Os arquivos alterados se restringiram ao *service*, onde estão as implementações das funções e ao DTO (Data Transfer Object), marcados em vermelho na figura 1b do conjunto de figuras 1, que são classes que definem quais objetos estão sendo transportados entre as camadas da aplicação.

No CF, foram criadas funções para salvar temporariamente as imagens enviadas durante a criação do caso. Em seguida, essas funções montam e executam o comando que aciona o CP para realizar a classificação. O resultado desse processo consiste em um texto informando a classificação da imagem e as imagens processadas, contendo marcações que indicam células lesionadas e não lesionadas.

Após a conclusão da classificação, as imagens temporárias são excluídas, enquanto as imagens geradas, juntamente com as informações fornecidas pelo autor do caso, são armazenadas, no banco de dados, para visualização pelos usuários. Para permitir tais funcionalidades, foi necessário utilizar a biblioteca *fs.promises*, [Node.js Foundation 2025], para salvar e excluir as imagens localmente, *child.process*, [Node.js Foundation 2025], para abir o terminal e inserir o comando para a execução do CP e *path*, [Node.js Foundation 2025] para construir os caminhos das imagens enviadas, caminho de salvamento das imagens que serão geradas e do arquivo de pesos do modelo IA treinado.

Os dados gerados pelo CP estarão disponíveis para visualização no aplicativo somente após interação dos usuários com o caso, com exceção do autor que poderá visualizar logo após a criação do caso e geração desses dados. Para permitir a visualização, foi criado um botão na tela de visualização de detalhes de caso, visto na figura 2a do conjunto de figuras 2, que navega para uma nova tela criada onde temos as imagens e o texto



(a) Tela de detalhes do caso após interação por voto ou visualizada pelo autor.

(b) Tela de visualização dodos gerados pelo classificador.

Figura 2. Tela de visualização de detalhes do caso e das informações geradas pelo classificador.

gerados pelo CP, como podemos ver na figura 2b do conjunto de figuras 2.

4. Resultados

Foi implementado o *back-end* utilizando NestJS e DTOs, resultando em um código mais modular, organizado e escalável. A separação em entidades e a centralização no *module* do *back-end* facilitam a manutenção, a troca de serviços externos e a melhoria da aplicação. Este último foi perceptível pelos autores durante a integração do classificador, ao ser necessário a modificação de somente dois arquivos para a integração do CP.

Mesmo com a alteração, o aplicativo manteve seu fluxo de informações, preservando funcionalidades como login, criação e visualização de casos, incluindo o registro de imagens, prazos e descrições, além do acompanhamento dos votos e resultados. A tela de visualização de resultados foi alterada, sendo adicionado um botão que permite a navegação para a tela de exibição da classificação, porém essa alteração não impactou o funcionamento e fluxo original.

5. Conclusão

O CF foi desenvolvido para apoiar citopatologistas no diagnóstico de lesões, promovendo colaboração e reduzindo erros. O *back-end* foi reformulado para permitir a integração de um classificador, mantendo a modularização, a organização e a estabilidade do aplicativo. A nova arquitetura facilita a manutenção e futuras atualizações sem comprometer o fluxo de informações.

A integração do classificador não impactou no fluxo original da aplicação, sendo necessário uma alteração mínima para que funcionasse. Com a adição do conteúdo gerado pelo classificador, contribuímos com o objetivo inicial do CF, visto que foi adicionado mais informações para que os PFs possam utilizar na construção do diagnóstico.

Referências

Diniz, D. N., Keller, B. N. S., Rezende, M. T., Bianchi, A. G. C., Carneiro, C. M., Oliveira, R. R. e. R., Luz, E. J. S., Ushizima, D. M., de Medeiros, F. N. S., and Souza, M. J. F. (2022). A cytopathologist eye assistant for cell screening. *AppliedMath*, 2(4):659–674.

Fundação Oswaldo Cruz (2023). Instituto nacional de saúde da mulher, da criança e do adolescente fernandes figueira. portal de boas práticas em saúde da mulher, da criança e do adolescente. postagens: Coleta e indicações para o exame citopatológico do colo uterino. Rio de Janeiro, 25 mai. 2023. Disponível em: `https://portaldeboaspraticas.iff.fiocruz.br/atencao-mulher/coleta-e-indicacoes-para-o-exame-citopatologico-do-colo-uterino/`. Acessado em: 01 out. 2024.

Gomes, M. F. d. S. L., Romano, S. M. V., and Dias, J. C. (2023). Modularização de aplicativos ios. *Revista Processando o Saber*, 15:01–15.

Guimarães, T. D. (2021). Desenvolvimento de um aplicativo para colaboração entre citopatologistas. Monografia (Graduação em Ciência da Computação) – Universidade Federal de Ouro Preto. Último acesso em 05 de setembro de 2024.

Iftikhar, M., Kandhro, I. A., Kausar, N., Kehar, A., Uddin, M., and Dandoush, A. (2024). Plant disease management: a fine-tuned enhanced cnn approach with mobile app integration for early detection and classification. *Artificial Intelligence Review*, 57(7):1–29.

Instituto Nacional do Câncer (2022). Conceito e magnitude. Instituto Nacional do Câncer. Último acesso 29 de julho de 2024.

Keller, B., Guimaraes, T. D., Malaquias, P. I. d. S., Ferreira, G. M. d. S., Resende, M. T., Carneiro, C. M., and Bianchi, A. G. (2021). Citofocus: Uma plataforma para colaboração e aprendizado em citopatologia. In *Anais do XXI Simpósio Brasileiro de Computação Aplicada à Saúde*, pages 404–409. SBC.

Node.js Foundation (2025). Node.js v20.0.0 documentation - child_process module. `https://nodejs.org/api/child_process.html`. Acessado em: 14 maio 2025.

Rehman, A.-u., Ali, N., Taj, I., Sajid, M., and Karimov, K. S. (2020). An automatic mass screening system for cervical cancer detection based on convolutional neural network. *Mathematical Problems in Engineering*, 2020(1):4864835.

Rezende, M. T., Bianchi, A. G., and Carneiro, C. M. (2021). Cervical cancer: Automation of pap test screening. *Diagnostic cytopathology*, 49(4):559–574.

World Health Organization (2024). Cervical cancer. World Health Organization. Disponível em: `https://www.who.int/news-room/fact-sheets/detail/cervical-cancer`. Último acesso em 03 de setembro de 2024.