AVALIAÇÃO DE MÉTODOS, TÉCNICAS E FERRAMENTAS DE DEVOPS PARA DESENVOLVIMENTO DE PORTAIS DE MONITORAMENTO DE EPIDEMIAS

Jose Pedro Daniel Dembo[1]
Tales Paiva Nogueira[2]

RESUMO

Este trabalho aborda o processo de transformação de um projeto de um website estático que apresenta dados sobre Covid-19 (casos confirmados, óbitos e vacinação) em um website dinâmico, aplicando metodologias ágeis baseadas em DevOps, como Integração Contínua. As abordagens usadas são apresentadas mostrando suas vantagens, assim como também as dificuldades enfrentadas durante a implementação. O funcionamento interno da aplicação consiste na realização de um processo de ETL (Extraction, Transformation, Load), tendo como destino dos dados um serviço de Cloud Storage. Os dados são extraídos de duas fontes, sendo uma via API do Governo Federal — OpenDataSus, e a outra pela coleta de dados do repositório o covidbr. Uma vez carregados, os dados são lidos pela aplicação web para a criação de gráficos e apresentação dos mesmos para os usuários finais. O objetivo deste trabalho é trazer comparações, mostrando os elementos positivos e negativos em relação a essa transição da arquitetura anterior da aplicação para a mais recente versão, bem como as ferramentas utilizadas durante esse processo e as suas metodologias.

Palavras-chave: devops; desenvolvimento web; covid-19.

UNILAB, IEDS, Discente, josedembo18@aluno.unilab.edu.br[1]
UNILAB, IEDS, Docente, tales@unilab.edu.br[2]

ISSN: 2447-6161

INTRODUÇÃO

Em 2020, o mundo enfrentou uma pandemia que devastou muitas vidas e deixou muitas sequelas, onde muitos governos e a população em geral não tinha preparação para lidar com aquele cenário. Em pesquisa realizada pela Universidade Federal do Espírito Santo em 2020, observou-se que “a pandemia de Covid-19 levou sete a cada dez pessoas a consumir notícias diariamente e a se manter atualizadas sobre os acontecimentos por meio da televisão” (BOND, 2020). A mesma pesquisa revelou que 65% dos participantes buscaram informações sobre a Covid-19 em sites jornalísticos e blogs. Nesse sentido, é notória a importância de prover veículos de informação de qualidade para a população em geral. Uma das maiores necessidades face à situação era o fornecimento de informações credíveis e atualizadas sobre a Covid-19 em geral, como casos confirmados, óbitos, bem como a análise da evolução desses números, surgindo assim a necessidade de centralizar informações. Nesse cenário, foi comum a criação de web sites com propósitos de informar a situação por meio de dashboards [Sathya et al. 2022]. O projeto foco deste trabalho consiste na produção de um website estático (em sua primeira versão) com visualizações sobre a Covid-19, que posteriormente evoluiu para uma aplicação dinâmica aplicando técnicas DevOps. Essa transição permitiu a inclusão de recursos interativos e uma experiência de usuário mais rica. Neste trabalho, relatamos o processo de evolução do website entre as duas versões. A definição de ”website estático”utilizada neste trabalho é a de uma aplicação cujo conteúdo apresentado em cada página não muda independentemente de quem for o usuário que estiver acessando. Além disso, um arquivo HTML único é enviado pelo servidor a cada acesso. Já um website dinâmico, no contexto deste trabalho, possui conteúdos personalizados para cada usuário, sendo necessária a montagem de uma resposta específica a cada requisição. Em geral, a construção de um website dinâmico é mais complexa tanto em recursos computacionais como em termos de maior esforço e expertise do time responsável. Com a evolução dos processos ágeis surgiram extensões e melhorias das mesmas. Uma das melhorias recentes, que afeta mesmo equipes ágeis, envolve a separação dos times de desenvolvimento e de operações. Essa separação tem potencial de atrasar entregas e dificultar a comunicação entre os times. Face a essa situação, surgiu a técnicas DevOps [Gokarna and Singh 2021, Masud et al. 2022], visando aproximar esses times, criando, assim, equipes multidisciplinares e com uma comunicação melhor e mais efetiva. A perspectiva é que, ao adotar práticas de DevOps, as entregas de novas funcionalidades sejam mais rápidas [Krief 2022]. A técnica DevOps consiste em diversas fases, sendo elas: i) planejamento e priorização de funcionalidades, ii) desenvolvimento, iii) integração contínua (CI, do inglês continuous integration), iv) implantação contínua (CD, do inglês continuous deploy) e v) monitoramento contínuo [Krief 2022]. A implementação da cultura DevOps não deve ser feita de uma forma brusca, mas sim paulatinamente, por ser um processo que traz maior complexidade ao ecossistema de ferramentas e exige maior expertise dos integrantes do time [Nayanajith and Wickramarachchi 2024]. Porém, uma vez implementadas as fases fundamentais, a execução das próximas fases se torna mais fácil [Gokarna and Singh 2021].

METODOLOGIA

A metodologia empregada durante o projeto demandou a mesclagem de técnicas de desenvolvimento de software ágil e metodologia científica via levantamento bibliográfico, seleção de trabalhos relacionados, identificação de gaps, definição de hipóteses e questões de pesquisa, assim como a realização de experimentos através da implementação de novas funcionalidades no website que serviu como plataforma de testes. As etapas de desenvolvimento do projeto após os levantamentos bibliográficos e as análises de projetos relacionados, foram todas baseadas em um website teste, que constituiu a transformação ou

ISSN: 2447-6161

mudança do mesmo de um site estático para dinâmico, adicionando algumas novas funcionalidades, fazendo assim com que a plataforma tivesse uma arquitetura mais flexível e propícia para implementação de práticas DevOps, como elementos de alta disponibilidade e tolerância a falhas, processos de Integração Contínua, entre outros fazendo assim uma distinção entre o site estático (primeira versão) e o dinâmico (segunda versão). Para o desenvolvimento dos projetos usou-se o Git para o versionamento de código e também o GitHub como repositório remoto do código. Na segunda versão, técnicas DevOps foram introduzidas com novas ferramentas e serviços. Uma das etapas para o desenvolvimento de aplicação que se manteve na sua versão inicial foi o processo de Extração dos Dados.

Em ambas as versões, três componentes principais permanecem: o usuário acessando o website, a aplicação em funcionamento e servindo os usuários finais e o processo de Extração, Transformação e Carregamento (ETL, do inglês Extraction, Transforming and Load), sendo executado e automatizado. A fonte de origem dos dados são arquivos CSV que contém os dados históricos (séries temporais) de casos e óbitos devidos à Covid-19 durante o estado de pandemia. Para a realização do processo de ETL foram usados scripts Python e a plataforma GitHub Actions para automação do processo de coleta de dados. Os dados são coletados do repositório covidbr [Cota 2020].

Durante o ciclo de desenvolvimento, o website (versão 2), foi implementado com princípio de Integração Contínua, onde algumas ferramentas já existentes na versão 1 foram utilizadas, porém, com novas funcionalidades. Também foram testadas novas ferramentas e feitas comparações para se chegar a um consenso, e optou-se por usar o GitHub Actions em comparação com as ferramentas do mercado, por exemplo, o Jenkins, muito pela sua simplicidade, por sua integração com o GitHub e também porque o mesmo supria os requisitos da aplicação. Para isso foram criados os processos de build e deploy da imagem da aplicação utilizando Docker. Docker é uma tecnologia de software que roda máquinas linux e windows com o objetivo de criar, gerenciar e orquestrar containers. E containers são ambientes isolados (processos) que compõem recursos necessários para a execução de um software.
A aplicação que, em sua primeira versão não tinha um servidor gerenciável, passou a ter um servidor para rodar a mesma, onde consegue suportar aplicações robustas. Juntamente com o Flask também foi criado um serviço de banco de dados para o cadastro dos usuários e um serviço de autenticação de usuários, ambos gerenciados pela plataforma Supabase. O processo de ETL, que visa obter dados pré-processados de qualidade, tem como destino um serviço de armazenamento de dados na nuvem. Para isso, existem vários provedores com serviços free tier (gratuitos de forma limitada), e dentro destas possibilidades, o provedor de computação em nuvem escolhido para as implementações dos serviços de armazenamento de dados na nuvem e computação (hospedagem da aplicação) foi o Oracle Cloud Infrastructure — OCI, pois o mesmo oferece uma gama de serviços gratuitos.

RESULTADOS E DISCUSSÃO

Como mencionado, a versão inicial no site consistiu em um website estático contendo a parte da camada de ETL e a aplicação web sendo resistente a mudanças. Já a nova versão é adequada a mudanças como evoluções ou até mesmo desacoplamento de serviços para posterior substituição por outros serviços, por exemplo, uma evolução na questão de conteinerização da aplicação uma vez que o processo de Integração Contínua constrói a imagem da aplicação. Estes elementos enfatizam ainda mais a vantagem da necessidade do processo de transformação do site de estático para dinâmico. A arquitetura inicial funciona na base de um

ISSN: 2447-6161

website estático hospedado no GitHub Pages, tendo como componentes principais os serviços do GitHub como o repositório remoto de código, o GitHub Actions (responsável pela automação de workflows que executa os processos de ETL dos dados assim como gera os arquivos HTML necessários), GitHub Pages (serviço responsável por carregar os arquivos HTML e disponibilizá-los em um website estático).

Figura 1. página inicial do website versão 2

A Figura 1 mostra a página inicial do website, onde se pode verificar os cards com informações de casos confirmados e óbitos. A versão atual conta com uma aplicação dinâmica, cuja construção passou por uma fase de mudança de ferramentas, processos, metodologias e hospedagem. O acesso à aplicação é feito via navegador, seja ele desktop ou mobile. A aplicação realiza a criação e carregamento dos cards e gráficos, fazendo a leitura dos dados a partir do armazenamento de dados na nuvem via protocolo específico, no caso OCI. É no serviço de armazenamento de dados na nuvem onde os dados pré-processados são destinados após a conclusão do processo de ETL, onde o processo é automatizado utilizando o GitHub Actions. A aplicação também se conecta com o banco de dados Postgres gerenciado pelo Supabase juntamente com o serviço de autenticação utilizando a biblioteca Python Supabase. Desta forma, os usuários podem se cadastrar no website, visualizar os seus perfis, e realizar a exclusão dos seus dados, caso queiram. O fato dos dados dos usuários estarem sendo armazenados é um potencial elemento para que uma evolução do website possa acontecer, como, por exemplo, personalização de visualização das informações e gráficos baseando-se na localização dos usuários, caso os mesmos forneçam suas informações de localização.

Um dos objetivos face à transformação do website foi fazer a arquitetura ser agnóstica a tecnologias de computação em nuvem ao máximo possível, dentro das necessidades e demandas da aplicação, e com isso temos a aplicação web feita em Python “conteinerizada”, onde a automação da criação das imagens é feita utilizando o GitHub Actions que realiza a build e o deploy da aplicação no repositório remoto do Docker — o Docker Hub. A aplicação web (Python e Flask) está sendo hospedada em um servidor na Oracle Cloud utilizando o seu serviço de computação.

É possível observar uma mudança no layout da página, porém mantendo sempre a estrutura principal, onde aparecem os cards primeiros depois os gráficos, e também o cabeçalho (header) da página contendo os mesmos elementos da primeira versão, porém temos agora um elemento diferente no cabeçalho, um link para fazer o login na página, que uma vez clicado nos dá acesso à página ilustrada na Figura 2.

Figura 2. Ilustração da página de login da aplicação

Na Figura 2 pode-se verificar a imagem que representa a página de login do usuário e na Figura 3, a página de cadastro de usuário. O login pode ser realizado via e-mail e senha ou utilizando uma conta Google. Pode-se verificar a página de cadastro do usuário que coleta dados como o nome completo e o nome de usuário.

Figura 3. Ilustração da página de cadastro da aplicação

CONCLUSÕES

A evolução da aplicação para sua versão dinâmica aplicando princípios das técnicas DevOps foi bem

ISSN: 2447-6161

sucedida, porém com vários pontos ainda a evoluir. A mudança foi significativa e causou um impacto aos usuários finais na utilização do website, principalmente os usuários da primeira versão, oferecendo uma melhor experiência do usuário, fator este que também pode ser melhorado com personalizações de visualizações de dados baseados na localização do próprio usuário, uma vez que agora a aplicação possui um sistema de cadastro de dados dos usuários. A aplicação possui atualmente uma arquitetura construída com um foco em facilitar mudanças, as evoluções e transformações futuras, caso sejam necessárias será de fácil adaptação ou execução, elementos como uma infraestrutura totalmente agnóstica a computação em nuvem onde uma aplicação conteinerizada é um ponto-chave para possibilitar isso. A implementação de técnicas de DevOps geram um grande diferencial no que diz respeito a resultados e facilidades de manutenção de serviços e automação de processos, aplicar a mesma não é fácil principalmente quando se tenta aplicar de uma vez só, as implementações exigem uma complexidade maior, tanto de domínio técnico como de também de ferramentas, porém esses elementos todos são recompensados com as diversas vantagens que oferecem para a aplicação em sua totalidade.

AGRADECIMENTOS

Os autores agradecem à UNILAB através do Programa PIBIC pelo apoio com a bolsa concedida ao estudante.

REFERÊNCIAS

BOND, Letycia. Pesquisa revela aumento do consumo de notícias durante pandemia. 2020. Disponível em: https://agenciabrasil.ebc.com.br/geral/noticia/2020-06/pesquisa-revela-aumento-do-consumo-de-noticias durante-pandemia. Acesso em: 06 out. 2021.

Cota, W. (2020). Monitoring the number of COVID-19 cases and deaths in brazil atmunicipal and federative units level. SciELOPreprints:362.

Decan, A., Mens, T., Mazrae, P. R., and Golzadeh, M. (2022). On the use of github actions in software development repositories. In 2022 IEEE International Conference on Software Maintenance and Evolution (ICSME), pages 235–245.

Gokarna, M. and Singh, R. (2021). Devops: A historical review and future works. In 2021 International Conference on Computing, Communication, and Intelligent Systems (IC-CCIS), pages 366–371.

Hammant, P. (2013). What is Trunk-Based Development? — paulhammant.com. https://paulhammant.com/2013/04/05/what-is-trunk-based-development/. [Acesso em 15-07-2024].

Krief, M. (2022). Learning DevOps: A comprehensive guide to accelerating DevOps culture adoption with Terraform, Azure DevOps, Kubernetes, and Jenkins. Packt Publishing Ltd.

Masud, S. M. R. A., Masnun, M., Sultana, A., Sultana, A., Ahmed, F., and Begum, N.(2022). Devops enabled agile: Combining agile and devops methodologies for software development. International Journal of Advanced Computer Science and Applications, 13(11).

Nayanajith, A. and Wickramarachchi, R. (2024). Challenges affecting the successful adoption of devops practices: A systematic literature review. In 2024 4th International

Conference on Advanced Research in Computing (ICARC), pages 311–315.

Sathya, K., Rao, P. M., and Babu, S. (2022). Development of a Covid-19 Information Dashboard to Access the Health Care Resources and Requirements Online. In 2022 3rd International Conference on Computing, Analytics and Networks (ICAN), pages 1–8.

ISSN: 2447-6161

ISSN: 2447-6161