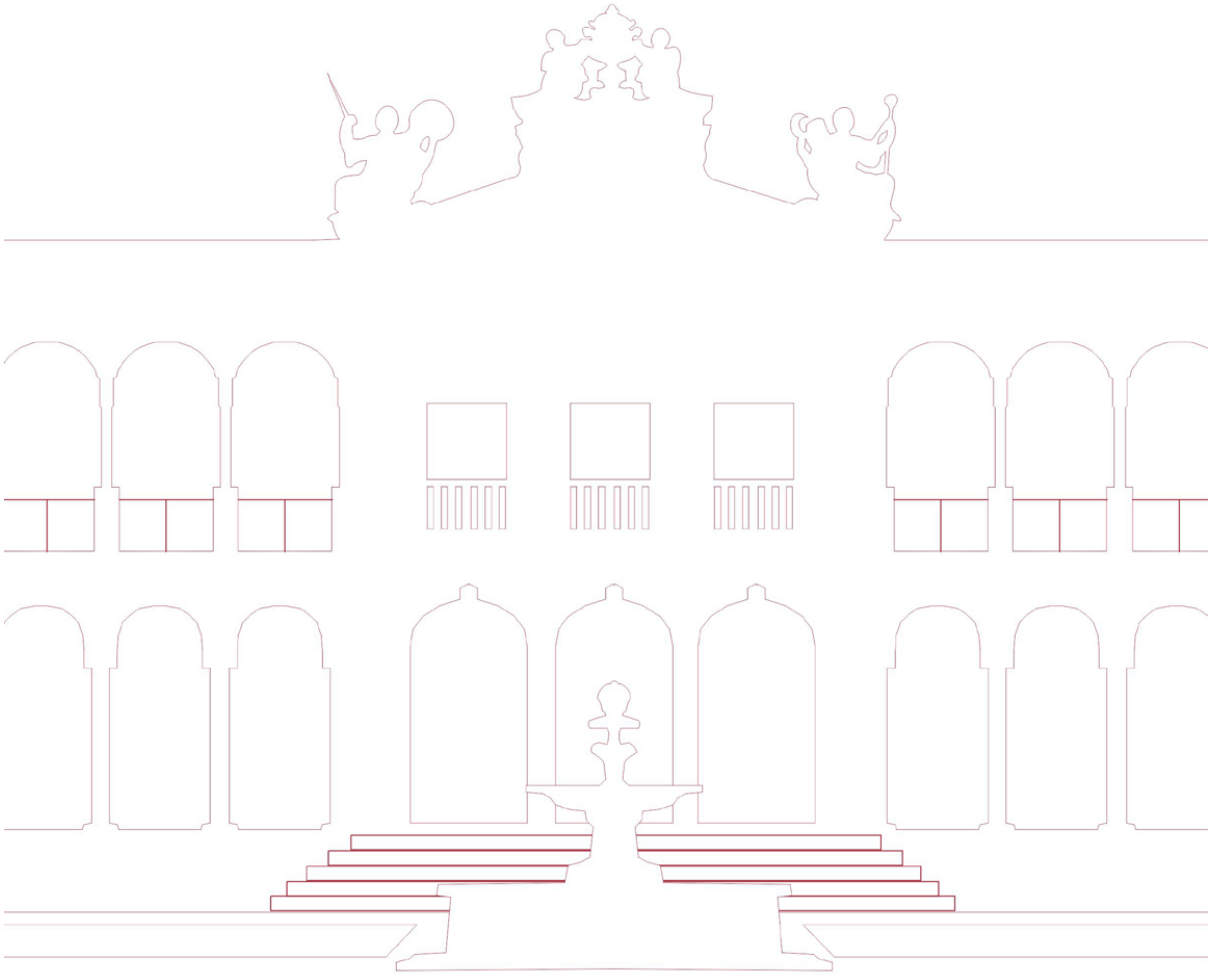
KPMG: Sapphirus DevOps

UNIVERSIDADE DE ÉVORA
ESCOLA DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA

Licenciatura em Eng. Informática
Estágio-Projecto 2022-2023



Carlos Palma (46520)

Orientador na empresa: João Simões
Orientador no departamento: Luís Rato

Trabalho desenvolvido na empresa KPMG no âmbito da disciplina de Estágio-Projeto da Licenciatura em Eng. Informática.

Évora, 5 de junho de 2023

Conteúdo

1 Introdução **1**
- 1.1 Enquadramento . . . 1
- 1.2 Objetivos . . . 1
- 1.3 Contribuições . . . 1
- 1.4 Estrutura do documento . . . 1

2 Ambiente empresarial **2**

3 Enquandramento teórico **3**

4 Estado da arte **4**

5 Ambiente de desenvolvimento **5**
- 5.1 Ambiente técnico . . . 5
  - 5.1.1 Hardware . . . 5
  - 5.1.2 Ferramentas de desenvolvimento . . . 5
  - 5.1.3 Ambiente aplicacional . . . 5
- 5.2 Metodologia de trabalho . . . 5

6 Trabalho desenvolvido **6**
- 6.1 Formação inicial . . . 6
- 6.2 Primeiro projeto . . . 6
- 6.3 Projeto final . . . 6
- 6.4 Design da solução . . . 6
- 6.5 Implementação . . . 7
- 6.6 Testes . . . 7
- 6.7 Documentação e suporte . . . 7

7 Avaliação crítica **9**

1 Introdução

O presente trabalho, realizado no âmbito da disciplina de Estágio, do plano de estudos da Licenciatura em Engenharia Informática, da Universidade de Évora, tem por base a experiência adquirida na empresa de estágio KPMG, realizado sob a orientação pedagógica de João Simões e supervisão do professor Luís Rato. Candidatei-me à vaga de DevOps no projeto Sapphirus da empresa KPMG, devido ao elevado número de tecnologias que iriam ser utilizadas considerei o estágio indicado para maximizar a minha aprendizagem. A KPMG é uma das principais empresas de consultoria, auditoria e serviços de contabilidade do mundo, com presença em mais de 150 países e o projeto Sapphirus, uma plataforma de dados baseada em nuvem que fornece análises avançadas e insights para ajudar as empresas a tomar decisões informadas. Enquanto DevOps fui responsável por ajudar na automatização de processos utilizando a plataforma GitHub, um DevOps é parte critica em qualquer projeto que envolva uma grande quantidade de membros, pois fornece à equipa capacidades de colaboração, integração contínua, entrega contínua e automatização, fazendo com que o projeto seja desenvolvido e entregue de forma mais rápida e eficiente.

Neste relatório apresento a minha experiência como DevOps destacando o conhecimento adquirido durante o estágio, ferramentas utilizadas, conceitos teóricos ou mesmo a forma como o trabalho em equipa é desenvolvido em grandes empresas como a KPMG.

1.1 Enquadramento

O estágio em questão foi realizado na empresa KPMG durante o período Fevereiro-Maio onde me encontrei integrado na equipa do projeto Sapphirus.

1.2 Objetivos

O estágio tinha como objetivo trabalhar num projeto focado em utilização de práticas de CI/CD. Trabalhei numa equipa que utilizava o GitLab e Github, Docker, Kubernetes, Java, JavaScript, Nexus e Hashicorp Vaul,tendo sido uma excelente oportunidade de adquirir experiência em práticas modernas de desenvolvimento e deployment de software em arquiteturas mistas event-driven, service-oriented, e decoupled-microservices.

1.3 Contribuições

Durante o meu estágio colaborei com a equipa de desenvolvimento para implementar, melhorar e manter as pipelines CI/CD, mais concretamente criando as pipelines CI/CD já existentes em GitLab, no GitHub. Ajudei a otimizar o processo de deployment da aplicação de forma automatizada para os vários ambientes, trabalhei com o Docker e o Kubernetes para containerizar e fazer deploy de aplicações e na gestão de artefactos no Nexus, incluindo o repositório Maven, o npm registry e o docker registry.

1.4 Estrutura do documento

Inicialmente irei enquadrar teóricamente os conceitos principais retratados ao longo do estágio e posteriormente irei abordar o trabalho desenvolvido com o decorrer das semanas. No fim realizarei uma apreciação critica a todo o trabalho desenvolvido.

2 Ambiente empresarial

A KPMG opera como uma rede global de firmas membro independentes, oferecendo serviços de audit, tax e advisory e trabalhando em estreita colaboração com os clientes, ajudando-os a mitigar os riscos e a aproveitar as oportunidades. Podemos encontrar firmas membro da KPMG em 143 países e territórios. Em conjunto, empregam cerca de 265.000 pessoas de diversas áreas. A KPMG em Portugal tem escritórios em Lisboa, no Porto e Évora com 68 membros da Partnership e cerca de 1300 colaboradores. [1]

A cultura empresarial da KPMG destaca-se pela busca constante de excelência e pela valorização do conhecimento e desenvolvimento dos seus profissionais. A empresa investe em programas de formação contínua e oferece oportunidades de crescimento e progressão na carreira. Além disso, a KPMG promove uma cultura de colaboração e trabalho em equipa, incentivando a troca de ideias e experiências entre os seus colaboradores. Os escritórios da KPMG em Lisboa, Porto e Évora refletem essa cultura empresarial dinâmica, são espaços modernos e bem equipados, projetados para incentivar a interação e a criatividade.

Na KPMG existem vários projetos, fiquei enquanto estagiário no projeto Sapphirus, no qual trabalha uma grande equipa sendo cada colaborador responsável por várias partes do desenvolvimento, desde a parte relativa à programação até outros aspetos como marketing e finanças. Fiquei inserido na equipa de DevOps constituída por mim e pelo meu orientador, o projeto é desenvolvido com uma metodologia de Integração/Entrega continua do tipo Scrum, todas as manhãs é realizada uma reunião em que os vários programadores fornecem feedback relativo aquilo que fizeram e vão fazer durante o dia, sendo este processo dividido em várias fases semanais.

A maior parte do trabalho foi realizado remotamente, tendo ido ao escritório em Lisboa apenas duas vezes, durante as minhas visitas ao escritório em Lisboa, pude observar o ambiente vibrante, onde profissionais de diferentes áreas se reúnem para discutir estratégias, partilhar conhecimentos e colaborar em projetos multidisciplinares.

3 Enquandramento teórico

Enquadramento Teórico: DevOps com GitHub

- DevOps/GitHub: Abordagem que busca a integração e colaboração contínuas entre as equipas de desenvolvimento e operações, visando acelerar o processo de desenvolvimento de software e melhorar a qualidade das entregas. O GitHub, uma plataforma popular para hospedagem e gestão de código-fonte, oferece recursos e ferramentas poderosas para a implementação de práticas DevOps eficazes. [4]
- Integração Contínua (CI) e Entrega Contínua (CD): A Integração Contínua refere-se à prática de integrar regularmente o código desenvolvido por diferentes membros da equipa, verificando a sua integridade através da execução automatizada de testes e análises estáticas. Isso permite identificar e corrigir problemas rapidamente, evitando conflitos e garantindo um código estável. A Entrega Contínua vai além, envolvendo a automação do processo de construção, testes e implantação, visando a entrega frequente e confiável de novas funcionalidades aos utilizadores. [5]
- Workflows: No GitHub, a configuração dos fluxos de trabalho para CI/CD é realizada através de ficheiros YAML. O YAML (Yet Another Markup Language) é uma linguagem de serialização de dados que permite definir de forma declarativa as etapas e tarefas necessárias para o processo de construção e implantação do software. Esses fluxos de trabalho consistem em jobs, que representam unidades de trabalho independentes, e steps, que são as tarefas individuais executadas em cada job. Os fluxos de trabalho podem ser personalizados para atender às necessidades específicas de cada projeto. [6]
- Actions: No contexto do GitHub, Actions são uma funcionalidade central para a automação de fluxos de trabalho. As Actions são componentes reutilizáveis que encapsulam um conjunto de etapas e tarefas comuns, permitindo a sua fácil integração nos fluxos de trabalho. Elas podem ser criadas pela comunidade ou personalizadas para atender às necessidades específicas do projeto. [7]
- GitHub Runner: A execução dos fluxos de trabalho é realizada pelo GitHub Runner. O GitHub Runner é um agente de execução que pode ser instalado em diferentes ambientes (como servidores, máquinas virtuais ou contentores) e que executa os jobs e steps definidos nos fluxos de trabalho. Ele permite a execução automatizada e controlada das tarefas definidas, garantindo a escalabilidade e a consistência na execução dos processos. [8]
- GitHub Secrets: Para garantir a segurança e a proteção de informações sensíveis, o GitHub disponibiliza a funcionalidade de secrets. Secrets são variáveis de ambiente criptografadas que podem ser utilizadas nos fluxos de trabalho para armazenar informações como senhas, chaves de API e tokens de acesso. Isso permite que essas informações sejam utilizadas durante a execução dos jobs sem expô-las diretamente nos ficheiros de configuração. [9]

4 Estado da arte

API Octokit vs API Github

A API do GitHub é a interface de programação de aplicações oficial fornecida pelo GitHub para interagir com os seus recursos e dados. Permite aos programadores aceder e manipular repositórios, problemas, pull requests, commits e outros recursos do GitHub. Por sua vez, a Octokit é uma biblioteca de cliente para a API do GitHub que simplifica e melhora a interação com a API. [2, 3]

Vantagens da utilização da biblioteca octokit:

- Abstração simplificada: A Octokit oferece uma camada de abstração que simplifica a interação com a API do GitHub. Fornece métodos e classes mais simples e intuitivos para realizar operações comuns, como criar um repositório, criar um problema ou enviar um pedido pull. Isto ajuda a reduzir a complexidade do código e acelera o desenvolvimento.
- Suporte a várias linguagens: A Octokit está disponível em várias linguagens de programação populares, como JavaScript, Ruby, Python e .NET. Isto permite que os programadores utilizem a Octokit na sua linguagem preferida, proporcionando uma experiência consistente em diferentes ambientes de desenvolvimento.
- Funcionalidades adicionais: A Octokit oferece recursos adicionais para além da API normal do GitHub. Por exemplo, pode lidar com autenticação de forma mais fácil, fornecendo métodos simples para autenticar pedidos à API. Além disso, a Octokit pode incluir funcionalidades específicas de cada linguagem, como manipulação de promessas em JavaScript ou gestão de tipos em TypeScript.
- Manutenção e suporte: A Octokit é mantida e suportada pela própria equipa do GitHub, o que significa que está sempre atualizada com as últimas alterações na API do GitHub. Isto garante que os programadores possam aproveitar rapidamente novos recursos e correções de erros, sem terem que lidar diretamente com a implementação da API.
- Comunidade ativa: A Octokit tem uma comunidade de programadores ativa, com recursos como documentação abrangente, exemplos de código e suporte técnico.

5 Ambiente de desenvolvimento

Nesta secção descrevo o ambiente de hardware e software onde trabalhei.

5.1 Ambiente técnico

5.1.1 Hardware

O computador que recebi para meu próprio uso era um portátil HP com um processador Intel Core i7 de oitava geração , 8 GB de memória e um disco rígido de 500 GB. Através desta máquina tinha acesso a uma VPN da Cisco que através de autenticação com recurso a username, PIN e password permitia-me aceder à rede da empresa e respetivos recursos, no meu caso acesso ao gitlab e github do projecto.

5.1.2 Ferramentas de desenvolvimento

A nível de software utilizei como ambiente de programação o Visual Studio Code que é o indicado pela empresa e uma Virtual Machine em Linux que me foi fornecida pela mesma. No âmbito do meu projeto fiquei responsável por criar as pipelines GitLab já existentes no projeto em GitHub pois a empresa adquiriu recentemente as tecnologias da Microsoft e pretende a longo prazo transacionar para as mesmas. Derivado deste meu trabalho principal fiquei responsável por fornecer a informação necessária aos meus superiores relativa aquilo que seria necessário para utilizar o GitHub no nosso projeto e como tal necessitei também de uma Virtual Machine para poder configurar o GitHub Runner responsável por correr os worflows criados por mim em GitHub.

5.1.3 Ambiente aplicacional

O meu trabalho estava inserido naquilo que é todo o processo de integração/entrega continua do projeto. Neste caso a empresa utilizava a plataforma GitLab e fui responsável por fazer a transição inicial para GitHub.

5.2 Metodologia de trabalho

No projeto em que estive inserido é utilizada uma metodologia de desenvolvimento de software do tipo Scrum, o trabalho é dividido em sprints e diariamente é realizada uma reunião para perceber em que ponto se encontra cada programador, qual o trabalho desenvolvido e o trabalho que vai ser desenvolvido. Existe um responsável pela delegação e planificação do trabalho a ser desenvolvido, sendo que cada programador tem depois acesso a uma ferramenta que permite indicar em que ponto se encontra o trabalho (em desenvolvimento, pronto para testes, entregue...). No meu caso específico, o meu trabalho no estágio diferiu um pouco dessa dinâmica. Fui designado para um único objetivo e reportava diretamente ao meu orientador, a quem eu apresentava o trabalho desenvolvido e o que planeava desenvolver nos dias seguintes.

6 Trabalho desenvolvido

Esta secção apresenta um resumo claro do trabalho desenvolvido durante o estágio, incluindo a lista de tarefas e respetivo timeline.

6.1 Formação inicial

Durante as primeiras duas semanas comecei por realizar várias formações relacionadas com aspectos de proteção de dados e segurança dentro da empresa que eram obrigatórias para poder prosseguir com o meu estágio dentro da KPMG. Após estas formações iniciais comecei a ser direcionado pelo meu orientador para algo relacionado com aquilo que viria a ser o meu trabalho. Durante o restante primeiro mês fiquei encarregue de aprender todos os conceitos relacionados com DevOps num paradigma de GitHub, comecei por procurar saber mais sobre aquilo que era o trabalho de um DevOps, de seguida o que era Continuos Integration e Continuos Deployment (senti que as aulas de MDS foram um grande auxilio nesta jornada) e por último aspetos mais técnicos como pipelines em GitHub, workflows, secrets, containers e a linguagem YAML. Ao longo deste processo foram-me sendo indicados pelo meu orientador locais onde obter tal informação, neste caso a documentação do GitHub e videos no youtube para entender os conceitos mais teóricos.

6.2 Primeiro projeto

Após o primeiro mês que foi unicamente constituído por formação foi-me então indicado o meu primeiro projeto prático de forma a aplicar o conhecimento obtido, neste primeiro projeto desenvolvi um workflow simples em que o objetivo era automatizar o processo de instalação dos pacotes necessários para uma aplicação Node.js aquando de um push para um certo repositório GitHub. O workflow começava por instalar o npm (gerenciador de pacotes Node.js), realizava o download dos pacotes necessários do repositório da empresa, instalava os mesmos e por fim realizava o release da aplicação no repositório da empresa. Consegui concluir este projeto muito mais rápido do que esperava (menos de uma semana) e mesmo assim considerei o mesmo bastante desafiante pois foi o meu primeiro contacto prático com um workflow e com a linguagem YAML. Neste primeiro projeto configurei também o GitHub Runner (responsável por correr os workflows que fui desenvolvendo), para tal utilizei uma Virtual Machine com o sistema operativo Linux. Sem dúvida que trabalhar com Linux na licenciatura me ajudou bastante pois foram inúmeros os problemas de permissões com que me deparei durante a configuração do Runner e já ter alguma experiência permitiu-me ultrapassar estes com alguma facilidade.

6.3 Projeto final

Concluído o primeiro projeto comecei então a desenvolver o projeto final, este consistia no desenvolvimento de parte da pipeline já existente em GitLab do projeto Sapphirus em GitHub, mais concretamente a parte relativa à verificação de commits, publicação de typescript library's e aplicações node.js. Este foi um trabalho realmente desafiante que me capacitou com várias competências, além de ter de criar todos os workflows em YAML e configurar novamente o GitHub Runner, trabalhei também com secrets, variáveis de ambiente, criei scripts em JavaScript e utilizei pedidos Rest. Este projeto que passarei a detalhar ocupou-me todo o restante tempo de estágio.

6.4 Design da solução

Comecei por tentar entender a pipeline já existente em GitLab, listar os secrets existentes e perceber quais teriam de ser as alterações a realizar tanto nos ficheiro YAML como JavaScript de forma a criar algo funcional em GitHub. Entendi então que parte dos scripts seriam iguais no entanto nos que era necessário alterar seriam inúmeras as alterações devido a utilizarem API'S que apenas existem em GitLab, passei então a procurar API'S em GitHub que fizessem aquilo que eu necessitava, o que me levou a ler bastante documentação. Em relação aos workflows podia manter os comandos bash utilizados em GitLab, no entanto toda a estrutura, condições e sintaxe teria de ser alterada de forma a ir ao encontro daquilo que é necessário em GitHub.

6.5 Implementação

Após a fase de design iniciei então a implementação da minha pipeline, inicialmente desenvolvi os meus scripts, optei por utilizar a API Octokit do GitHub para realizar os pedidos Rest ao GitHub e receber no meu script as informações necessárias, neste caso precisava de uma lista de todos os commits realizados desde a última tag criada, em GitLab existia um pedido que me retornava exatamente isto, no entanto em GitHub, tive de andar a "brincar"com os pedidos e objetos JSON retornados, enviando e recebendo informação de forma a obter o pretendido. Após ter um objeto com todos os commits desde a última tag criei então outro script que era responsável por verificar se existiam commits bem formatados e quais (segundo as normas do projeto) , passei então a desenvolver a primeira parte do meu workflow, tendo em conta que existiam várias partes de código que teria de repetir durante todo este, procurei uma solução para que pudesse chamar o código tal como acontece com uma função em Java por exemplo (de forma a melhorar a leitura e manutenção do mesmo), após ler documentação entendi que o mais adequado seria criar actions com as partes do código que iria repetir e chama-las sempre que necessário. Desenvolvi então actions para a instalação do npm e download dos vários pacotes do repositório na empresa, neste caso era necessário realizar o download a partir de vários links daí a maior dificuldade que no primeiro projeto. Criados estes primeiros jobs que eram responsáveis apenas por instalações e downloads pude então começar a criar os jobs mais difíceis, primeiramente criei o job "commit-checker"que verificava se se tratava de um push ou pull e de seguida utilizava os scripts que desenvolvi (enviando para o mesmo a informação do push/pull) para verificar se existiam commits bem formatados e apenas em caso positivo o wokflow continuava a sua execução, guardando o nome do último commit bem formatado numa variável de ambiente. O objetivo deste workflow é receber as typescript library's criadas, testar as mesmas, de seguida criar um nome e versão caso ainda não se encontrem definidos e por fim publicar a mesma no nexus da empresa (repositório centralizado com as bibliotecas, pacotes etc), como tal de seguida criei o job "test" que simplesmente com recurso a um script verificava se a biblioteca estava de acordo com as normas da empresa e bem formatada. Finalizada a fase de teste criei então o "create-image-name-helper"que é responsável por verificar se a biblioteca já tem nome e versão definidas e em caso negativo com recurso a mais alguns scripts criar o nome e respetiva versão. Após todo este processo passei então à criação do job final "release"que foi provavelmente o mais fácil de desenvolver, consistindo apenas no envio da library para o nexus da empresa. Todo este processo está aqui apenas resumido pois foram inúmeras as dificuldades encontradas em cada passo do desenvolvimento, no GitHub as funcionalidades são bastante diferentes do GitLab, inicialmente estava a desenvolver o meu trabalho em 3 workflows separados mas devido as dificuldades em relacionar workflows em GitHub, tive a necessidade de desenvolver todo o projeto em apenas um workflow. Todos os jobs apresentam dependências entre eles e utilizam variáveis de ambiente para passar a informação, apenas quando um job é concluído com sucesso é possível passar para o próximo. De forma aos meus scripts poderem aceder ao meu repositório e realizarem pedidos foi necessário definir um GITHUB-TOKEN de forma a dar acesso privilegiado aos scripts. Foi necessário definir vários secrets para guardar informação privada que não deve ser acessível por todos os membros do projeto.

6.6 Testes

Para testar o meu workflow no GitHub, utilizei o GitHub Runner como um executor de tarefas, permitindo a execução automatizada e contínua dos jobs. A configuração e a definição dos jobs foram realizadas através de ficheiros YAML, de acordo com as práticas recomendadas do GitHub.

Após a criação de cada job, iniciei os testes para validar seu funcionamento e integração adequados. Utilizei casos de teste específicos para verificar se as ações definidas em cada job estavam a ser executadas corretamente. Além disso, garanti que os artefactos produzidos por cada job fossem armazenados corretamente e que a comunicação entre os diferentes jobs e etapas do fluxo de trabalho estivesse a funcionando adequadamente.

O uso do GitHub Runner mostrou-se altamente confiável, fornecendo uma infraestrutura robusta para a execução automatizada dos jobs. A comunicação entre o GitHub e o GitHub Runner foi estável, garantindo uma sincronização eficiente entre as alterações no repositório e a execução dos jobs.

6.7 Documentação e suporte

Durante o desenvolvimento do projeto, a documentação desempenhou um papel fundamental, e a transição para o GitHub facilitou essa tarefa, uma vez que o projeto já estava hospedado no GitLab. A

utilização do fluxo de trabalho para os membros da equipa foi praticamente a mesma tanto no GitLab quanto no GitHub, o que permitiu que eu mantivesse toda a documentação existente, com apenas algumas alterações em relação ao funcionamento dos scripts. A documentação foi atualizada para refletir as mudanças feitas no fluxo de trabalho do GitHub, foram adicionadas instruções claras sobre como utilizar os recursos específicos oferecidos pelo GitHub, como a criação de issues, pull requests e a gestão dos workflows. Utilizei ferramentas de documentação, como Markdown, para criar e manter a documentação atualizada.

7 Avaliação crítica

Considero desde já este estágio como uma das experiência mais enriquecedoras naquilo que é a minha jornada enquanto aluno e futuro engenheiro informático e a minha ideia inicial e a razão porque escolhi este estágio revelou-se correta. A nível técnico melhorei o meu conhecimento sobre a forma como funciona o Git, aprendi a trabalhar com scripts e a desenvolve-los nas mais variadas linguagens tendo sido inúmeros os conceitos teóricos que aprofundei e aprendi com esta experiência. Fiquei estupefacto e entusiasmado com a forma como o trabalho é desenvolvido em grandes equipas de programadores, ajudou bastante o facto de a empresa me ter permitido assistir todas as manhãs de estágio às reuniões diárias realizadas pela equipa. Muitos dos conceitos já tinham sido abordados na disciplina de metodologias de desenvolvimento de software no entanto ter um contacto real com aquilo que acontece sem dúvida que motiva bastante e teria ajudado ter esta experiência aquando da ocorrência da disciplina. Relativamente à forma como o trabalho foi desenvolvido considero que poderia ter realizado os testes de outra forma pois fiz alguma pesquisa durante o estágio e existem ferramentas que permitem facilitar o teste de workflows (Travis CI), no entanto foi algo que acabei por não implementar na minha metodologia de trabalho. Foi também possível ter um pequeno vislumbre daquilo que quero fazer no futuro, pois tive contacto com pessoas que trabalhavam nas mais variadas áreas e trabalhei com várias tecnologias, preferi sem dúvida o trabalho que realizei com recurso a JS e acredito que na minha segunda experiência profissional irei engendrar pela área do BackEnd. As minhas competências de trabalho em equipa e comunicação acredito que não tiveram um grande desenvolvimento tendo em conta que ainda antes da maioridade já trabalhava em cafés, restaurantes e bares onde adquiri estas previamente. Tanto o meu orientador da empresa como o da universidade foram fantásticos, sendo que o João me acompanhou durante todo o meu percurso auxiliando sempre que eu tinha alguma dúvida ou problema no projeto e o professor Luís Rato que se preocupou em realizar a reunião para verificar se tudo estava a correr conforme planeado para o meu estágio. Em suma aprendi bastante do ponto de vista prático durante estes últimos meses e acredito que esta experiência terá um papel preponderante naquilo que é o meu futuro, foi possível entender que aquilo que aprendi na universidade tem uma grande utilidade e aplicabilidade no dia a dia.

Referências

[1] KPMG. *About KPMG*. Disponível em: https://kpmg.com/pt/pt/home/about.html. Acesso em: 31 de maio de 2023.

[2] . *Octokit*. Disponível em: https://github.com/octokit. Acesso em: 31 de maio de 2023.

[3] . *API Github*. Disponível em: https://docs.github.com/en/rest?apiVersion=2022-11-28. Acesso em: 31 de maio de 2023.

[4] . *DevOps*. Disponível em: https://azure.microsoft.com/pt-pt/resources/cloud-computing-dictionary/what-is-devops. Acesso em: 1 de Junho de 2023.

[5] . *Entrega e Integração continuas*. Disponível em: https://unity.com/pt/solutions/what-is-ci-cd. Acesso em: 1 de Junho de 2023.

[6] . *GitHub workflows*. Disponível em: https://docs.github.com/en/actions/using-workflows. Acesso em: 1 de Junho de 2023.

[7] . *GitHub Actions*. Disponível em: https://github.com/features/actions. Acesso em: 1 de Junho de 2023.

[8] . *GitHub Runner*. Disponível em: https://docs.github.com/pt/actions/using-github-hosted-runners/about-github-hosted-runners. Acesso em: 1 de Junho de 2023.

[9] . *GitHub Secrets*. Disponível em: https://docs.github.com/pt/actions/security-guides/encrypted-secrets. Acesso em: 1 de Junho de 2023.